

Flixan Utilities

The Flixan program includes additional utilities that perform other type of functions that further refine, modify, augment and reformat the flight vehicle models. There is a utility for combining systems and matrices together using an algorithm that is not vulnerable to algebraic looping. Matlab often causes algebraic loops errors when closing a loop around a system. There are also utilities for combining transfer-functions into systems, transforming continuous systems to discrete using (z) or Tustin transformations with variable delays, utilities for transforming systems and matrices from a system files (.Qdr) to formats that can be loaded into Matlab or converting Matlab generated data to system files. There are also utilities that modify or reduce already existing systems. Modifying the inputs and outputs, or reducing the variables of a system by selecting a smaller number of inputs, states, and outputs, or modifying the system units by rescaling some of the variables. Additional outputs can also be included from the states, state derivatives, or output derivatives. These utilities are useful for separating coupled systems to longitudinal and lateral subsystems. There are also utilities for processing the input data in batch mode. Batch instructions call multiple other utilities, process the entire input file, and generate systems without any interactive assistance from the user.

Each Flixan utility reads and processes a set of instructions data which are included in the input file (.Inp), in a similar fashion as the vehicle, the actuator and other Flixan programs. An input data file typically contains multiple data-sets to be processed by different utility programs. Each data-sets is processed by the corresponding utility program which typically creates a new system in the systems file according to the instructions and the data read from the set. The input data-sets are created either by using a text editor, copied from another file and modified, or more efficiently, each utility program itself is capable of generating its own input set interactively with the user, by means of menus and dialogs, as we shall see in the examples.

Initially the analyst must generate the desired dynamic models and matrices for his analysis and must prepare the input file by setting-up and testing each data-set, one set at a time. The dynamic models created may be rigid-body or high order models with flexibility, actuators, TWD, slosh, etc. They may include flight control or sensor dynamics. They may be coupled or decoupled, with pitch and lateral separated. They may be open-loop or closed-loop depending on the analysis required. When all the elements in the input file are prepared and tested for one flight condition, the modeling becomes easier and faster for the next run or case, because the same input file can be modified and be reused for a different situation or iteration. A batch can be included which can instantly processes the entire input file.

1.0 Systems Combination Program

The systems combination program is a Flixan utility that interconnects several state-space systems together and with matrices, which are read from a systems file (.Qdr). The combined system created is in state-space form and it is saved in the same systems file. The program uses two data files: an input data file (.Inp) and the systems file. The input file contains the titles of the systems to be combined, the interconnection instructions, and the definitions of input and output variables. The interconnection instructions describe how the various subsystems are combined together and how they connect to the big system inputs and outputs. There is also an interactive Graphics/ User interface (GUI) utility included that helps the user to enter the interconnection data in the input file or to read the interconnections from the file graphically without editing the file directly. Figure (1-1) shows the interaction between the systems combination program, the GUI, the user, and the data files.

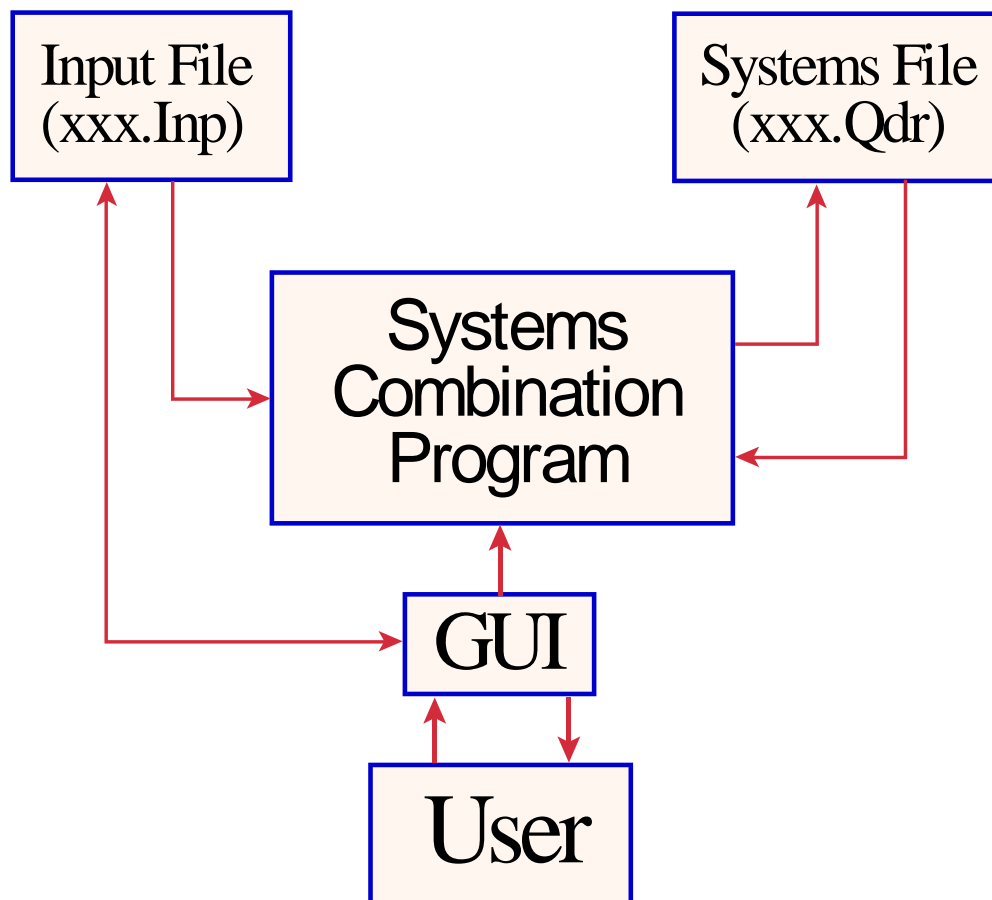


Figure (1.1) Diagram of the Systems Combination Utility Program showing its interaction with the User and the Input/ Output Files

1.1 Program and Input/ Output Files Selection

The systems combination program is selected from the Flixan main menu by clicking on the “*Program Functions*”, select the “*Creating and Modifying Linear Systems*” group, and then click on “*Combine State-Space Systems and Matrices*”. Then use the following file selection dialog to select the input data file and the systems file from the menus, as shown below.



Systems Interconnection data-set in the input file (.Inp)

System interconnection data-sets contain the system interconnection instructions and they are located in input files (.Inp). A typical data-set for combining two systems is shown below.

```
INTERCONNECTION OF SYSTEMS .....
RULV, Pitch Closed-Loop, Simple Model
! Combine the Simple Vehicle model with the Actuator model in Closed-Loop
! Form for Time-Domain Simulation and Check the Eigenvalues
Titles of Systems to be Combined
Title 1 RULV, Pitch Axis, Max_Q (T=85 sec)
Title 2 Main Engine Actuator
SYSTEM INPUTS TO SUBSYSTEM 2                               Eng. Actuator
System Input 1 to Subsystem 2, Input 1, Gain=-1.8          Theta_command
.....
SYSTEM INPUTS TO SUBSYSTEM 1                               Vehicl Distrib
System Input 2 to Subsystem 1, Input 7, Gain= 1.0          Gust Velocity
.....
SYSTEM OUTPUTS FROM SUBSYSTEM 1                           From Vehicle
Via Matrix +I6
.....
SYSTEM OUTPUTS FROM SUBSYSTEM 2                           From Actuator
System Output 7 from Subsystem 2, Output 1, Gain= 1.0     Delta_Engine
System Output 8 from Subsystem 2, Output 2, Gain= 1.0     Delta_Dot_Eng
.....
SUBSYSTEM NO 2 GOES TO SUBSYSTEM NO 1                    Actuat to Veh
Subsystem 2, Output 1 to Subsystem 1, Input 1, Gain= 0.4082 Eng.Defl # 1
Subsystem 2, Output 1 to Subsystem 1, Input 2, Gain= 0.4082 Eng.Defl # 2
Subsystem 2, Output 1 to Subsystem 1, Input 3, Gain= 0.4082 Eng.Defl # 3
Subsystem 2, Output 1 to Subsystem 1, Input 4, Gain= 0.4082 Eng.Defl # 4
Subsystem 2, Output 1 to Subsystem 1, Input 5, Gain= 0.4082 Eng.Defl # 5
Subsystem 2, Output 1 to Subsystem 1, Input 6, Gain= 0.4082 Eng.Defl # 6
.....
SUBSYSTEM NO 1 GOES TO SUBSYSTEM NO 2                    Vehi to Actu
Subsystem 1, Output 1 to Subsystem 2, Input 1, Gain= 1.8   Theta
Subsystem 1, Output 2 to Subsystem 2, Input 1, Gain= 2.3   q - rate
Subsystem 1, Output 6 to Subsystem 2, Input 1, Gain=-0.008 Nz Accelerat
.....
Definitions of Inputs = 2
Attitude Theta Command (radians)
Wind Gust Disturbance (ft/sec)

Definitions of Outputs = 8
Pitch Attitude (theta-rigid) (radians)
Pitch Rate (q -rigid) (rad/sec)
Angle of attack (alpha) (radians)
Change in Altitude (dh) (feet)
Forward Acceleration (V-dot) (ft/sec)
Accelerometer # 1, (along z), (ft/sec^2)
Actuator Delta_Engine (radians)
Actuator Delta_Dot_Engine (rad/sec)
-----
```

The first line in the data-set "INTERCONNECTION OF SYSTEMS", defines the type of input data and the utility program that will process it. This is because the input file typically also contains data for other Flixan programs, such as, transfer function combination data, flight vehicle modeling data, etc. The second line contains the title of the new system to be created. Every system in Flixan is identified by a title which is less than 100 characters long. The following few lines, starting with (!), are comment lines. They provide a short description of the new system for documentation purposes and they are optional. The comment lines are also copied in the systems file below the new system title. The lines below the line "Titles of Systems to be Combined", contain the titles of systems to be combined together. These systems must already exist in the systems file (.Qdr), otherwise, an error will occur. Notice that each system is assigned a reference number, (1, 2, 3, ... max= 9). The program searches the systems file (*.Qdr), finds the subsystem titles, reads the quadruple data for each

```

BATCH MODE INSTRUCTIONS .....
Systems Interconnection Testing Batch
! A Batch Set used for generating and connecting systems together for demo purposes.
! It first retains two matrices in the systems file that will be used later in the connections.
! Then it generates two second order systems from Transf-Functions. Then a second order with
! 1 sec delay. Then it combines the two 2nd order systems together to create a 4th order system.
! Then it combines two identical 4th order systems in series to create an 8th order system.
! Then it combines four of these 8th order systems in parallel to create a 32nd order system.
! Then it creates a 14th order system by combining 3 of the above systems (2nd, 4th, and 8th).
! Then we create a system by combining two different 2nd order Transf-Functions in parallel.
! We also create a big system of 116 states, 5 inputs and 5 outputs by combining seven of the
! above systems together. We also combine the 14th order with the 8th order systems via gain
! matrix connections to create a 22nd order system. We finally create a 5th order system using
! transf-functions and combine the 5th order with a 4th order and also an 8th order systems to
! create another system.
Retain Matrix      : (3x2) Gain Connection matrix
Retain Matrix      : (2x2) Gain Connection matrix
Transf-Functions   : (s^2 +2s +16)/(s^2 +5s +25)
Transf-Functions   : 4/(s^2 +2s +4)
Transf-Functions   : 4/(s^2 +2s +4)*exp(-sT)
System Connection: Fourth Order System
System Connection: Eighth Order System
System Connection: 32nd Order
System Connection: Fourteenth Order System (3 Systems Combined)
System Connection: Two Second Order Transfer Functions Connected in Parallel
System Connection: Two Systems in Series (12th Order)
System Connection: A Big System
System Connection: Two Systems Connected with a Gain Matrix
Transf-Functions   : 4/(s^2 +2s +4)*exp(-sT) (b)
Transf-Functions   : Fifth Order System of 3 Transfer Functions
System Connection: Another System Combination

```

```

-----
SYSTEM OF TRANSFER FUNCTIONS ...
(s^2 +2s +16)/(s^2 +5s +25)
! A simple 2nd order numerator 2nd order denominator transfer function
Continuous
TF. Block # 1 Second order, (2 zeros/2 poles)                Order of Numer, Denom=  2  2
Numer 1.00000      2.00000      16.0000
Denom 1.00000      5.00000      25.0000
-----
Block #, from Input #, Gain
  1  1      1.00000
.....
Outpt #, from Block #, Gain
  1  1      1.00000
.....
Definitions of Inputs = 1
Transf. Function (s^2 +2s +16)/(s^2 +5s +25) Input #1

Definitions of Outputs = 1
Transf. Function (s^2 +2s +16)/(s^2 +5s +25) Output #1
-----

```

```

SYSTEM OF TRANSFER FUNCTIONS ...
4/(s^2 +2s +4)
! Another second order denominator transfer function
Continuous
TF. Block # 1 Low Pass Filter (2 rad/sec bandwidth)          Order of Numer, Denom=  0  2
Numer 0.0          0.0          4.0
Denom 1.00000      2.00000      4.00000
-----
Block #, from Input #, Gain
  1  1      1.00000
.....
Outpt #, from Block #, Gain
  1  1      1.00000
.....
Definitions of Inputs = 1
Transf. Function 4/(s^2 +2s +4) Input # 1

Definitions of Outputs = 1
Transf. Function 4/(s^2 +2s +4) Output # 1
-----

```

SYSTEM OF TRANSFER FUNCTIONS ...

4/(s^2 +2s +4)*exp(-sT) (b)

! This is a second order low pass filter with a cut-off frequency of 2 (rad/sec) in series with a ! one second Pade delay approximation.

Continuous

TF. Block # 1 Low Pass Filter (2 rad/sec bandwidth) Order of Numer, Denom= 0 2

Numer 0.0 0.0 4.0

Denom 1.00000 2.00000 4.00000

TF. Block # 2 Delay (sec)= 1.00000 Order of Numer, Denom= 1 1

Outpt #, from Input #, Gain {Combined System Outp} from {Low Pass Filter Inpu}

Block #, from Input #, Gain {TF: 1, Low Pass Fil} from {Low Pass Filter Inpu}

Outpt #, from Block #, Gain {Combined System Outp} from {TF: 2, Delay (sec)=}

Block #, from Block #, Gain {TF: 2, Delay (sec)=} from {TF: 1, Low Pass Fil}

Definitions of Inputs = 1

Low Pass Filter Input

Definitions of Outputs = 1

Combined System Output

SYSTEM OF TRANSFER FUNCTIONS ...

Fifth Order System of 3 Transfer Functions

! This System consists of three transfer functions in feedback configuration

Continuous

TF. Block # 1 Second Order (a) Order of Numer, Denom= 2 2

Numer 1.0 1.0 3.0

Denom 1.0 2.0 5.0

TF. Block # 2 Second Order (b) Order of Numer, Denom= 2 2

Numer 1.0 0.5 2.0

Denom 1.0 4.0 6.0

TF. Block # 3 First Order Order of Numer, Denom= 1 1

Numer 1.0 1.0

Denom 1.0 4.0

Block #, from Input #, Gain {TF: 1, Second Order} from {Combined System Inpu}

Outpt #, from Block #, Gain {Combined System Outp} from {TF: 2, Second Order}

Block #, from Block #, Gain {TF: 2, Second Order} from {TF: 1, Second Order}

Definitions of Inputs = 2

Combined System Input 1

Combined System Input 2

Definitions of Outputs = 2

Combined System Output 1

Combined System Output 2

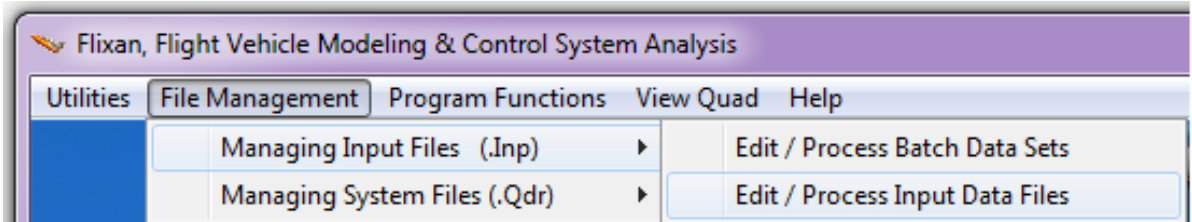
```

INTERCONNECTION OF SYSTEMS .....
Another System Combination
! Combine three systems together from above
!
Titles of Systems to be Combined
Title 1 Fourth Order System
Title 2 Eighth Order System
Title 3 Fifth Order System of 3 Transfer Functions
SYSTEM INPUTS DIRECT TO SYSTEM OUTPUTS
System Input 1 to System Outpt 2, Gain= 0.10000
System Input 2 to System Outpt 1, Gain= 0.20000
.....
SYSTEM INPUTS TO SUBSYSTEM 1      Fourth Order System
System Input 1 to Subsystem 1, Input 2, Gain= 1.0000
System Input 2 to Subsystem 1, Input 1, Gain= 1.3400
.....
SYSTEM INPUTS TO SUBSYSTEM 2      Eighth Order System
Via Matrix +I2                      Size ( 2 x 2)
.....
SYSTEM OUTPUTS FROM SUBSYSTEM 2   Eighth Order System
System Output 2 from Subsystem 2, Output 1, Gain= 2.0000
System Output 1 from Subsystem 2, Output 2, Gain= 1.5670
.....
SUBSYSTEM NO 1 GOES TO SUBSYSTEM NO 2
Subsystem 1, Output 1 to Subsystem 2, Input 1, Gain= 2.0000
Subsystem 1, Output 1 to Subsystem 2, Input 2, Gain= 2.3456
Subsystem 1, Output 2 to Subsystem 2, Input 2, Gain= 1.2345
.....
SUBSYSTEM NO 2 GOES TO SUBSYSTEM NO 3
Via Matrix -I2                      Size ( 2 x 2)
.....
SUBSYSTEM NO 3 GOES TO SUBSYSTEM NO 1
Via Matrix -I2                      Size ( 2 x 2)
.....
Definitions of Inputs = 2
Combined System Input 1
Combined System Input 2

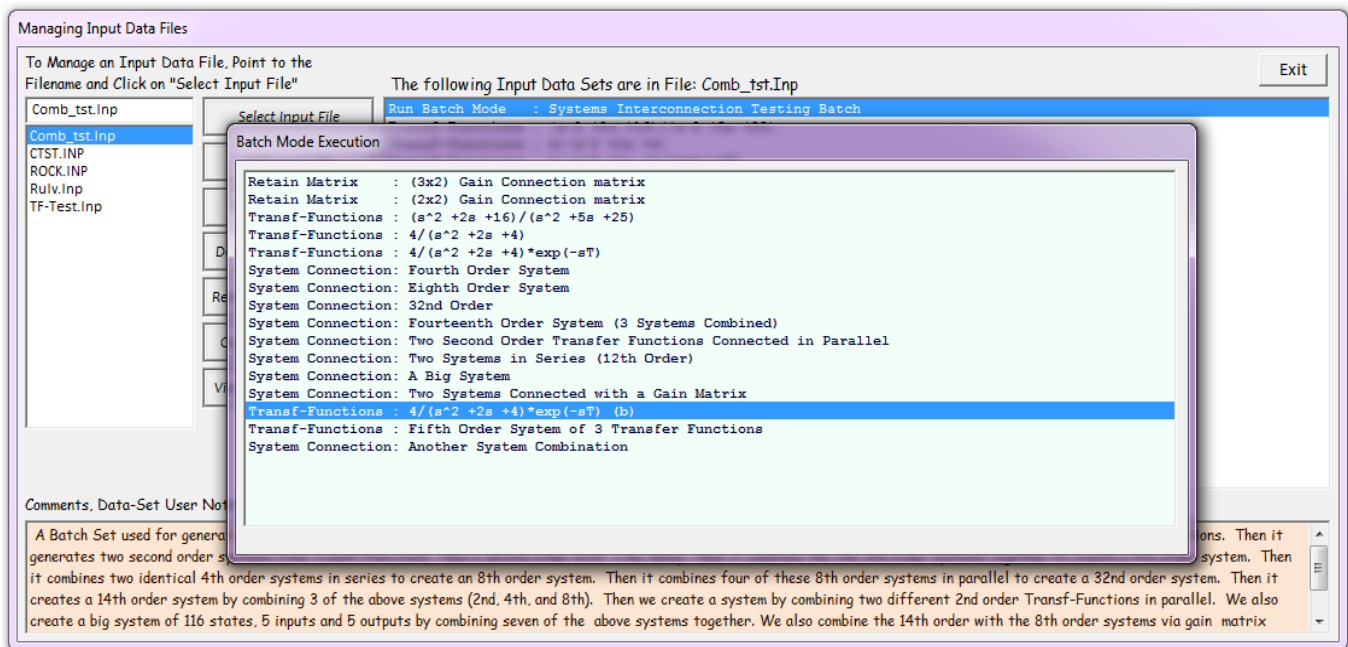
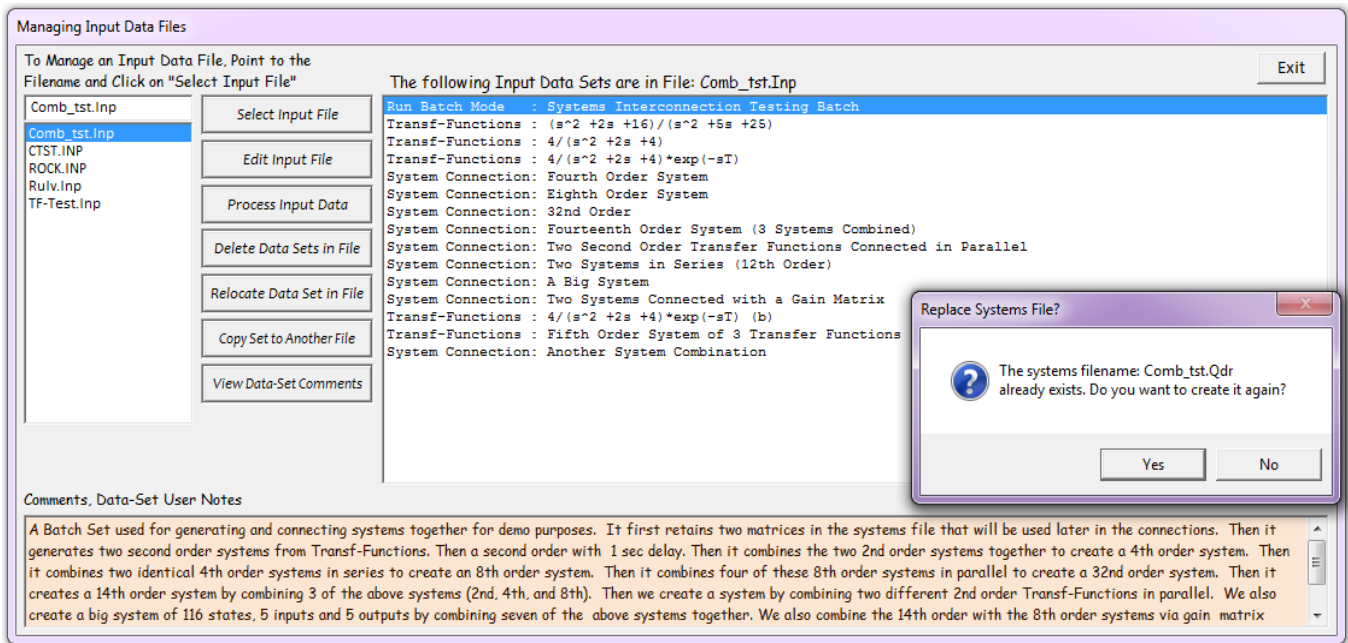
Definitions of Outputs = 2
Combined System Output 1
Combined System Output 2

```

Let us now process this file by running the batch set which is located on the top of the input file. First select the directory, and from the Flixan main menu select "File Management", then "Manage Input Files", and then "Edit/ Process Input Data Files", as shown.



The input data file manager dialog comes up that has two menus. From the left menu select the input file "Comb_tst.Inp" and click on "Select Input File". The menu on the right side shows the titles of the data-sets located in this file, also the program that will process each set. They are mainly system and transfer-function interconnection sets. Select the first title which is the batch set and it will process the remaining data-sets in batch (non-interactively). Click on "Process Input Data", and the batch set will process the input file and save the systems in file " Comb_tst.Qdr".



A display pops up temporarily while the batch is executing that shows the batch commands, as shown above. The currently executing command is highlighted for debugging purposes in case of errors. Notice that the batch includes two commands to retain and copy two matrices from the original file to the new file because the matrices are used in the interconnections. Otherwise, the old data in the systems file are erased and replaced with new data.

2.0 Transfer Functions Combination Program

The transfer functions combination program is a Flixan utility that can be used to combine several transfer function blocks together in a state-space system form. The combined state-space system is saved in a systems file. The program uses two data files, an input data file (.Inp) and a systems file (.Qdr). The input data file includes the transfer functions to be combined, the transfer function interconnections, and the definitions of the combined system's input and output variables. The interconnection instructions describe how the transfer functions are to be connected together by the program, and to the system inputs and outputs. The combined TF system is saved in the systems file. There is also an interactive Graphics/ User interface (GUI) utility included that helps the user to enter the transfer-function interconnection data in the input file or to read the interconnections from the file graphically without editing the file directly. Figure (2.1) shows the interaction between the transfer functions combination program, the GUI, the user, and the data files.

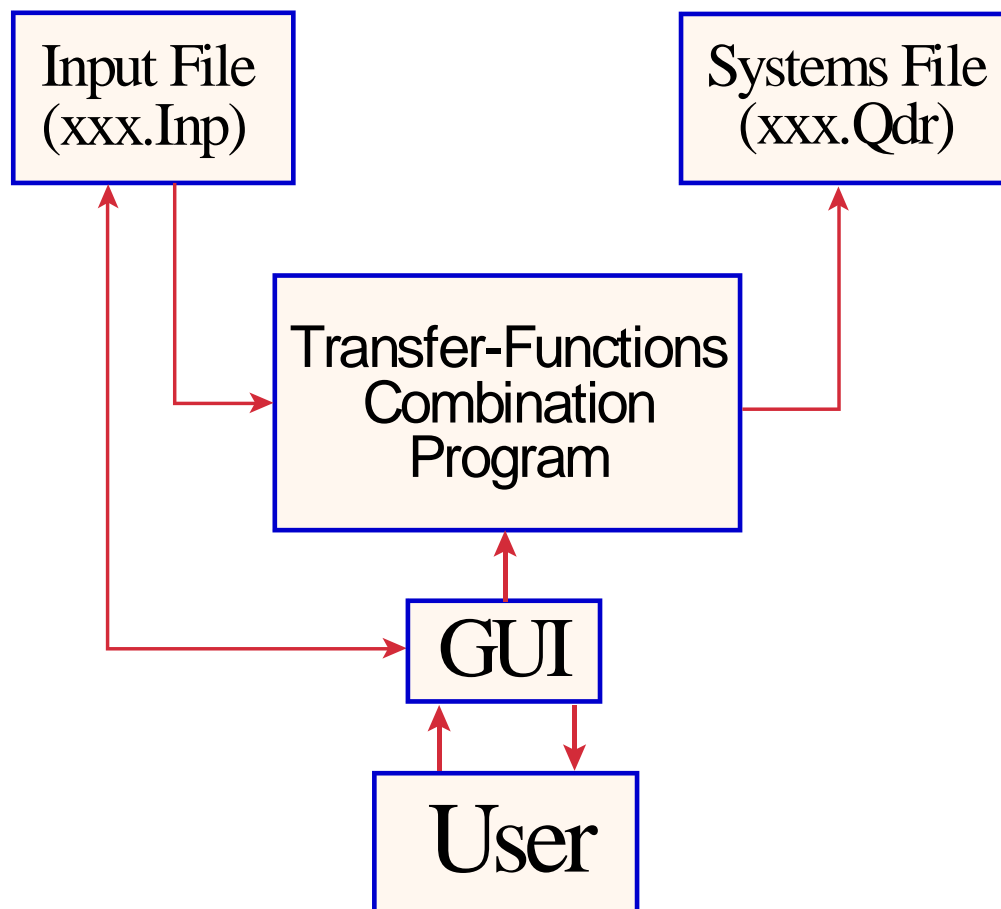


Figure (2.1) Diagram of the Transfer-Functions Interconnection Utility Program showing its interaction with the User and the Input/ Output Files

2.1 Program and Input/ Output Files Selection

The transfer functions combination program can be selected from the Flixan main menu by clicking on the “*Program Functions*”, select the “*Creating and Modifying Linear Systems*” group, and then click on “*Combine Transfer-Functions to State Systems*”. Then use the following file selection dialog to select the input data file and the systems file from the menus, as shown below. The two files are selected from a filenames selection menu.



Transfer Functions Interconnection data-set in the input file (.Inp)

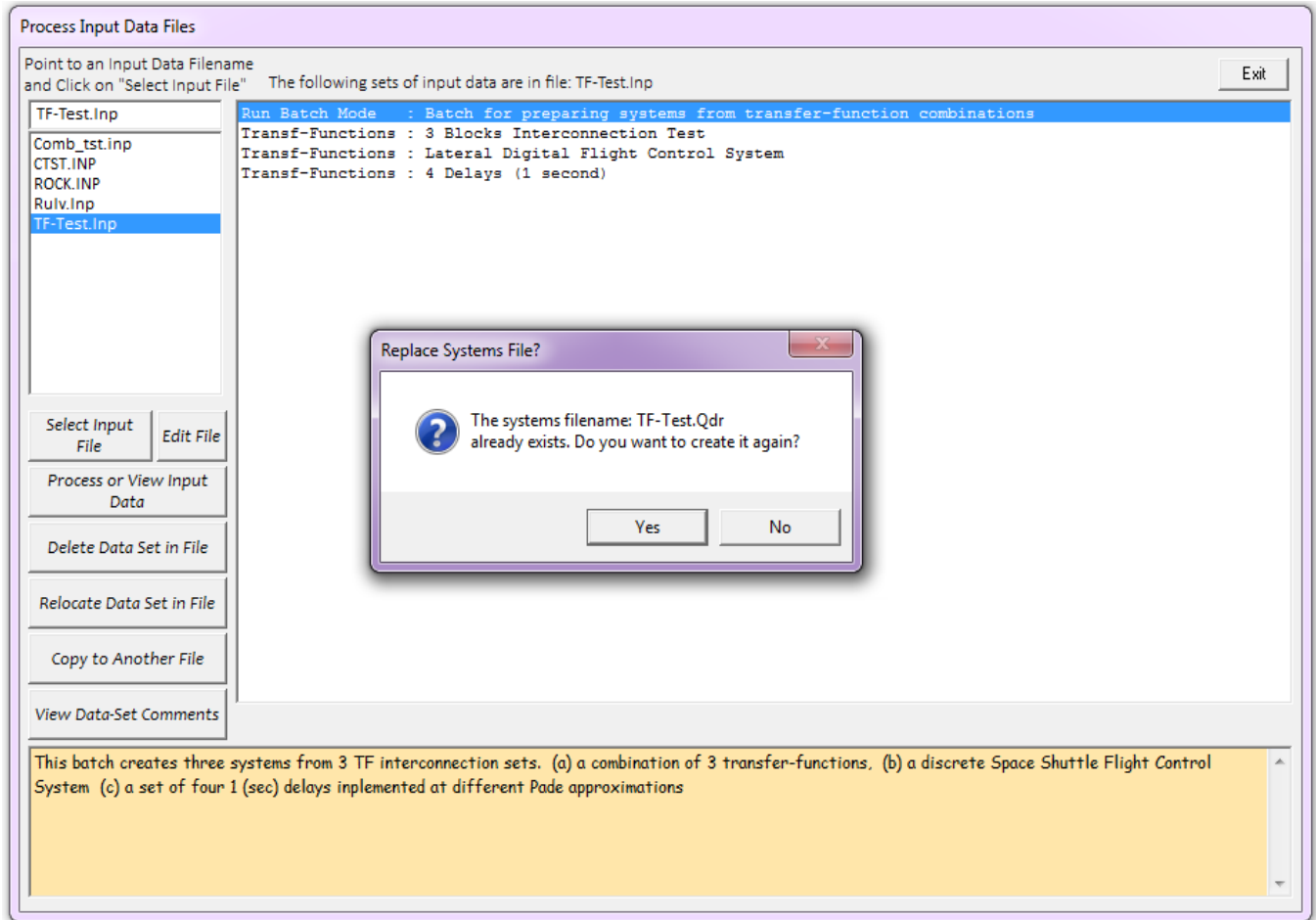
TF interconnection data-sets contain the transfer functions interconnection instructions and they are located in input files (.Inp). A typical data-set for interconnecting three transfer function blocks together is shown below.

```
SYSTEM OF TRANSFER FUNCTIONS ...
3 Blocks Interconnection Test
! Comment lines for the new system
! This system is obtained by combining 3 TF blocks
!
Continuous
TF. Block # 1 (s2+3s=2)/(s2+2s+0.75)           Order of Numer, Denom=  2  2
Numer 1.00000    3.00000    2.00000
Denom 1.00000    2.00000    0.750000
TF. Block # 2 1/(s+5)                         Order of Numer, Denom=  0  1
Numer 0.00000    1.00000
Denom 1.00000    5.00000
TF. Block # 3 (s3+5s2+6.75s+2.25)/(s3+3.4s2+3.2s+0.8)  Order of Numer, Denom=  3  3
Numer 1.00000    5.00000    6.75000    2.25000
Denom 1.00000    3.40000    3.20000    0.800000
-----
Block #, from Input #, Gain
   1         2         1.00000
   2         1         1.00000
   3         3         1.00000
.....
Block #, from Block #, Gain
   2         1         0.37500
.....
Output #, from Block #, Gain
   1         2         5.00000
   2         3         0.35555
   2         1         0.133333
.....
Output #, from Input #, Gain
   1         1         0.001
   2         1         0.355555
.....
Definitions of Inputs =  3
System Input 1 to Block 2
System Input 2 to Block 1
System Input 3 to Block 3
.....
Definitions of Outputs =  2
System Output 1 from Block 2
System Output 2 from Blocks 1 and 3
-----
```

At the top of the data-set there is a line describing the type of data following in order to identify the program that will process the data. In this case it is a system of transfer-functions interconnection. This is because the input file does not only contain transfer functions combination data for this program but it may also contain data-set for other Flixan programs (such as, for example, systems combination data, flight vehicle modeling, actuator, etc). In the second line we have the title for the new system to be created. It will not be a transfer function but it will be in state-space form. The next few lines below the title beginning with (!) are comment lines for documentation purposes which describe the new system and they are optional. The comment lines will be copied in the systems file below the new system title.

The next line defines if the transfer functions and subsequently the new system to be created are “Continuous” (s-plane), or “Discrete” (z-plane). If the TF are discrete the sampling period is also included in (seconds). For example, “Discrete, $dT= 0.04$ ”. The following lines define the transfer function blocks. Each block is assigned a TF block number and a short description title. The numerator and denominator orders of each TF are shown at the right end of the line. Below each TF title, in the

This example file can also be processed in batch mode by running the batch set which is located on the top of the input file "TF-Test.Inp", as shown below. From the Flixan main menu select "Edit", then "Manage Input Files" and then "Process/ Edit Input Data". The following dialog is used for managing and processing input files. Use the left menu to select the input file "TF-Test.Inp" and click on "Select Input File". The menu on the right side shows the titles of the data-sets located in this file, also the program that will process each set. The top set is a batch file that will process the remaining three TF sets together in batch, and not interactively. Click on "Process or View Input Data", and the batch set will process the data, and save the three systems in file "TF-Test.Qdr". The previous systems in this file will be replaced. More transfer-function and system combination examples are located in folder: "C:\Flixan\Comb_Trfn\Examples".



3.0 Systems Reconstruction Program

This is a utility program used to modify, reduce, simplify, separate or reconstruct an existing state-space system. For example, the analyst may wish to replace the system outputs with a new set of outputs, or change the units of the system variables, or to eliminate some unnecessary inputs, states, or outputs, or to separate pitch and lateral dynamics. The System Modification program reads the state-space system to be modified from a systems file ".Qdr" and creates new systems by modifying the original system's variables or may perform some of the following operations:

- Rescale the original system variables by converting the units of the old system's inputs, state, and output variables into different units. The conversion is defined by a constant scaling factor (S), where the old units = (S) times the new units.
- The utility can also create a new system by replacing the old system outputs with a new set of outputs which are made up from a combination of some of the old system's outputs, the states, and the derivatives of the old system outputs and states. This option is useful for extracting derivatives of outputs which are often needed when constructing control design models for LQG or H-infinity synthesis, where state or output derivatives are often included in the criteria optimization vector.
- It is also used to simplify a system by eliminating some of the unused old system's inputs, states, or outputs. It can also be used to decouple a state-space system. Separating, for example, a coupled flight vehicle model into pitch and a lateral subsystems. This option can also be used to change the sequence of the system's variables.

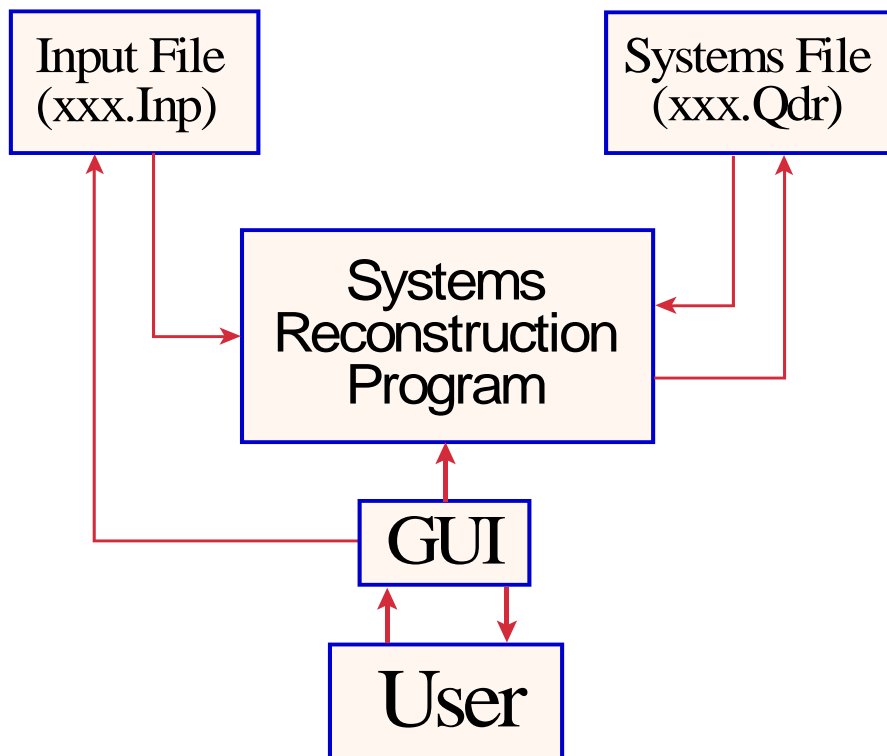
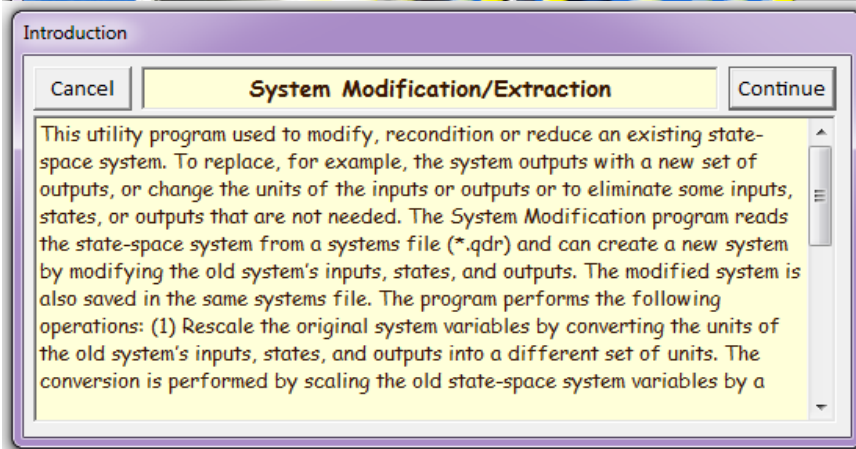
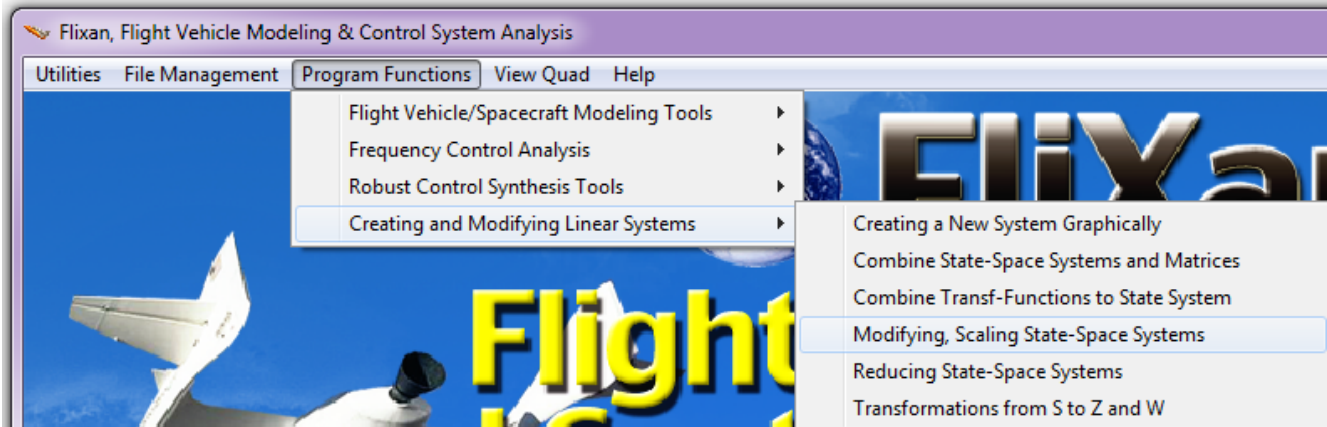
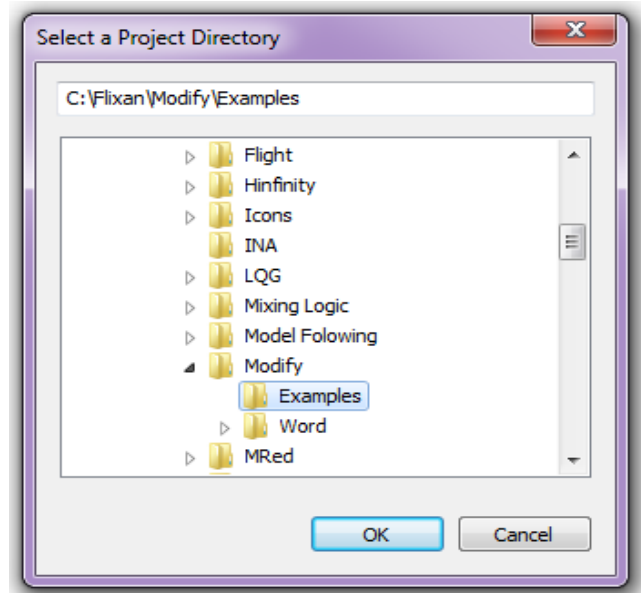
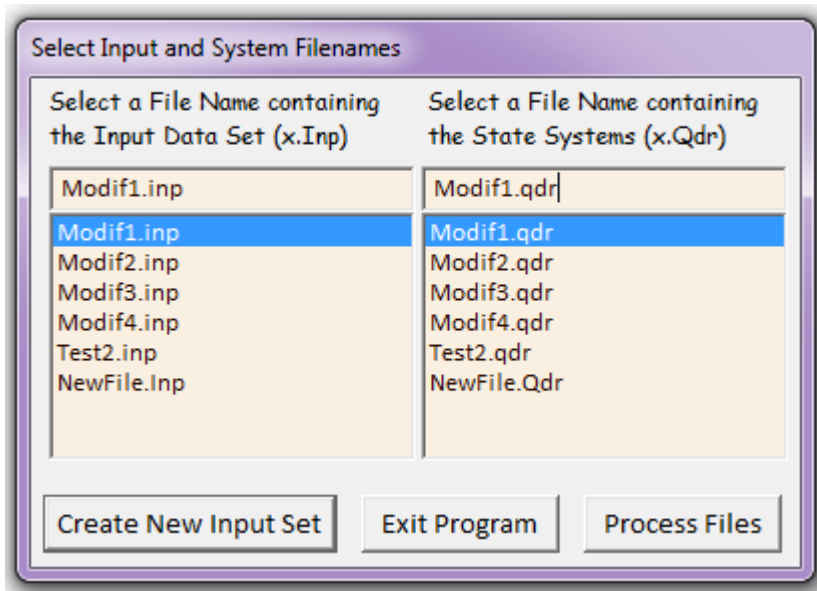


Figure (3.1) Diagram of the Systems Reconstruction Program used to Modify/ Rescale existing systems. It shows the interaction between with the User, Program, and Input/ Output Files

3.1 Program and Files Selection

The example files presented in this section are located in folder "C:\Flixan\Modify\Examples". The Systems Reconstruction/ Modification program can be selected from the Flixan main menu by clicking on "Program Functions", then select the "Creating and Modifying Linear Systems" group, and then click on "Modifying & Scaling State-Space Systems". Then using the following file selection dialog select the input data file and the systems file from the menus, as shown below. The input file contains the system reconstruction instructions, that is, when the instructions set has already been saved there from a previous run. Otherwise, the user must define the system modification instructions interactively which will be saved in file ".Inp" for a later use. The systems file originally contains the state-space system that will be modified by the program according to the instructions in the input set, such as, rescaling, selecting variables, creating new outputs, etc. The modified system will also be placed in the same systems file. After selecting the two filenames the user must click on "Process Files" to run the program, or if the instructions are not in file click on "Create New Input Set" to enter the system modification instructions interactively.



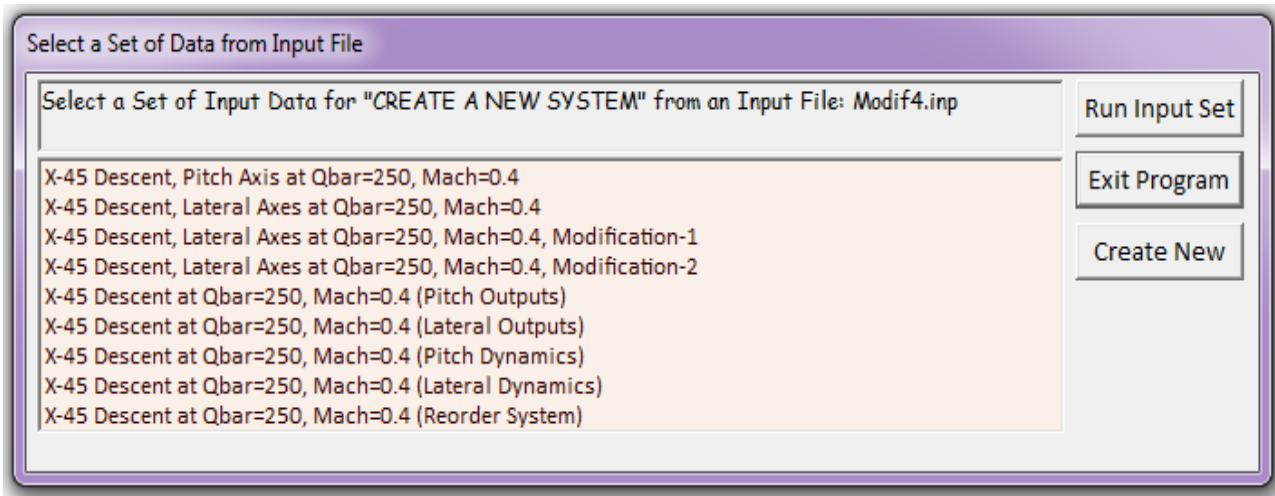


3.2 Instructions Already Saved in Input Data File

When the system modification data is already defined in the input file, the user selects the input and systems filenames from the menu above and clicks on "*Process Files*". The input file typically contains several sets of system modification instructions. Three typical system modification data-sets from file "*Modif4.Inp*" are shown below. They all begin with the line: "CREATE A NEW SYSTEM FROM AN OLD SYSTEM" which identifies the type of data and process, followed by system titles and instructions. There are two system titles: (a) the title of the modified new system to be created, and (b) the original system title before the modifications.

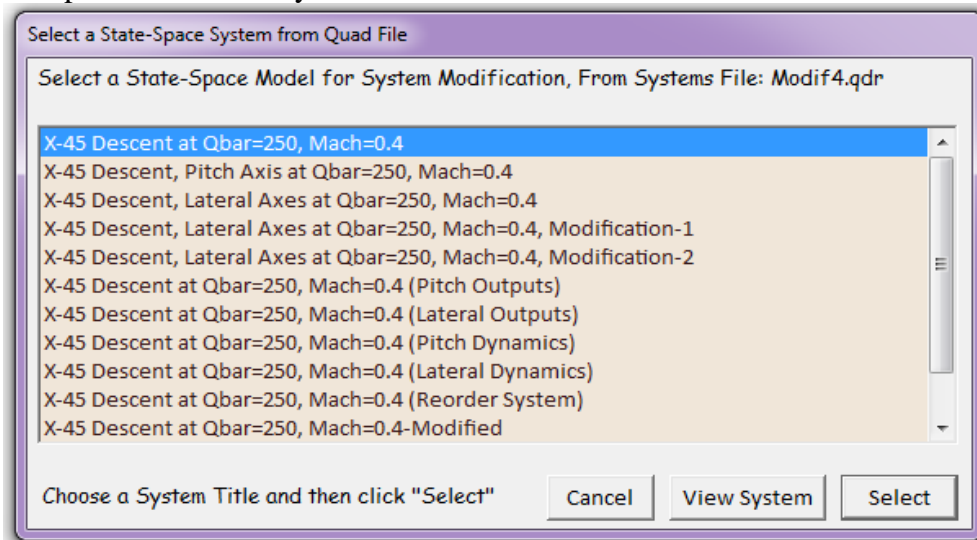
```
-----
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
X-45 Descent at Qbar=250, Mach=0.4 (Lateral Outputs)
X-45 Descent at Qbar=250, Mach=0.4
! From the original system, create a new set of outputs only for the lateral axis
NEW SYSTEM OUTPUTS FROM PREVIOUS SYSTEM OUTPUTS/ STATES/ DERIVATIVES
From Old System Output: 1 Roll Attitude (phi-Stabil) (radians)
From Old System Output: 2 Roll Rate ( p -Stabil) (rad/sec)
From Old System State : 5 Yaw Attitude (psi-rigid) (radians)
From Old System State : 6 Yaw Rate ( r -rigid) (rad/sec)
From Old System Output: 8 Angle of sideslip (beta-rigid) (radians)
From State Derivative : 8 Angle of sideslip (beta-rigid) (radians) /Rate
From Old System Output: 12 CG Acceleration along Y axis, (ft/sec^2)
From Output Derivative: 2 Roll Rate ( p -Stabil) (rad/sec) /Rate
From Output Derivative: 6 Yaw Rate ( r -Stabil) (rad/sec) /Rate
-----
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
X-45 Descent at Qbar=250, Mach=0.4 (Pitch Dynamics)
X-45 Descent at Qbar=250, Mach=0.4
! Extract the Pitch Dynamic Model from system: " X-45 Descent at Qbar=250, Mach=0.4"
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs : 1 4 5
Extract States : 3 4
Extract Outputs: 3 4
-----
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
X-45 Descent at Qbar=250, Mach=0.4 (Lateral Dynamics)
X-45 Descent at Qbar=250, Mach=0.4
! Extract the Lateral Dynamic Model from system: " X-45 Descent at Qbar=250, Mach=0.4"
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs : 2 3 5
Extract States : 1 2 5 6 8
Extract Outputs: 1 2 5 6 8 12
-----
```

The program searches through the input data file "Modif4.Inp" looking for system modification data-sets. When it finds them it displays their titles in a menu, shown below, and in this case it found many system modification sets in this file. These are the first titles that define the new system to be created and not the original system titles. The user selects one of the new system titles from this menu and clicks on "Run Input Set" to process it. The program reads the input instructions and calculates the modified state-space system from the original system. Then it saves the new system in the systems file "Modif4.Qdr" using the new system title selected from the menu.

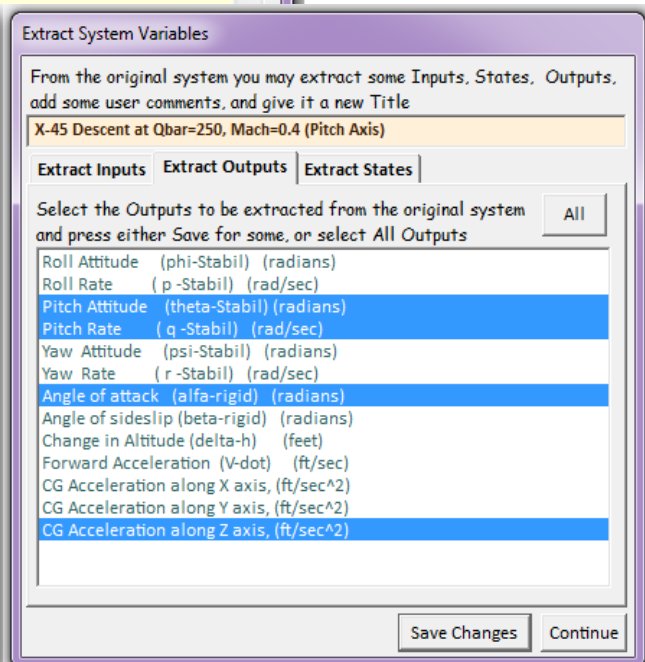
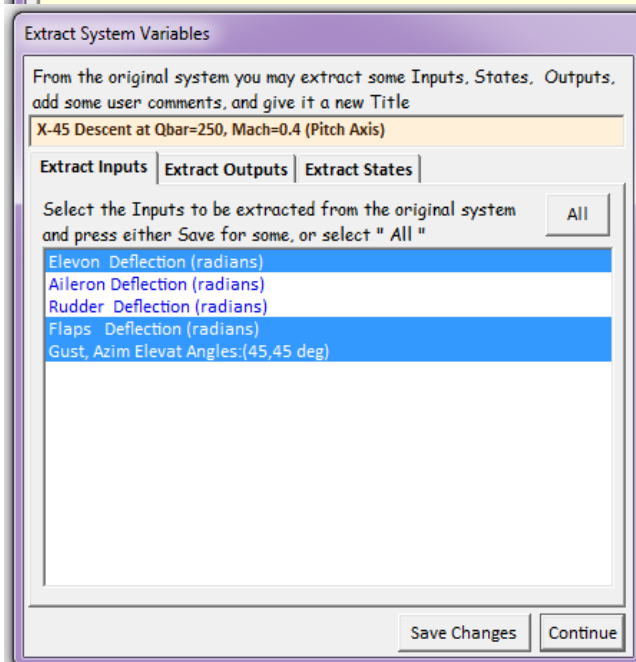
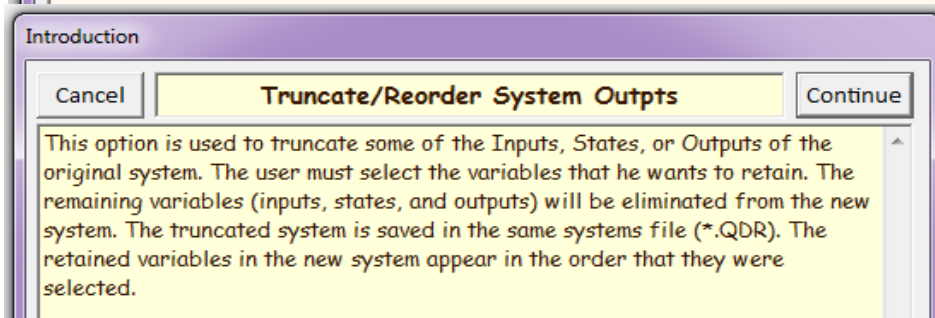
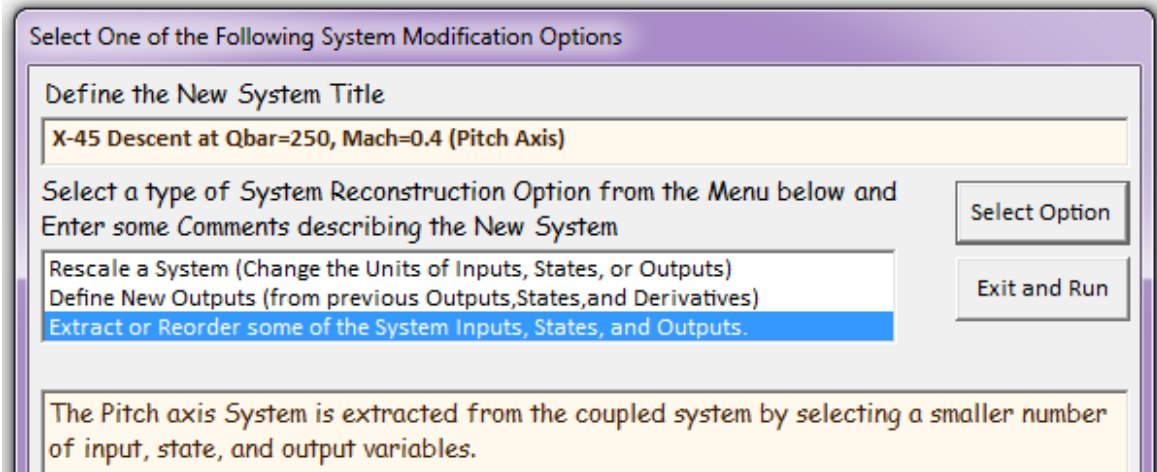


3.3 Creating a New Set of System Modification Instructions

If the system modification instructions are not in file (.Inp) the program will create the instructions set interactively, save it in file (.Inp), and then process it. After selecting the input and system files from the filename selection menu the user must click on "Create a New Input Set" instead of "Process Files" as in the previous section. If the input filename does not exist the user may enter a new name with an extension (.Inp). The program will receive the system modifications interactively, save them as a data-set in the input file and process the set. The following menu shows the titles of the systems included in file "Modif4.Qdr". Select the first one to be modified. It is a vehicle model that includes both pitch and lateral dynamics.



In this example we will create a new system by extracting only the pitch variables from the original system. Using the following dialog enter the new system's title "X-45 Descent at $Q_{bar}=250$, $Mach=0.4$ (Pitch Axis)". From the menu select the third option "Extract or Reorder some of the System Variables", enter some comments below describing the new system, and click on "Select Option" to continue. The following dialog shows the definitions of the original system's inputs, states, and outputs in three separate menu tabs. From the menus the user selects the variables to be retained in the new system. Click on "Save Changes" every time you modify the choices. Then click on "Continue" to save and process the new set of system modification data.



3.4 Creating a New Set of System Rescaling Instructions

This option is used for converting the units of some of the system inputs, states, or output variables from one type of units to another. Converting, for example, a system whose inputs are defined in pounds and its outputs in feet to a dynamically equivalent system whose inputs are defined in kilograms and its outputs are in meters. It is used, for example, to transform a plant model in order to combine it with a controller or an actuator systems that is defined in different units. In order to scale a state-space variable from one set of units to another the user must define the conversion factor “S”. This scaling factor is defined as: one old unit is equal to “S” times the new unit.

Let us consider, for example, a simple second order system that models the response of a spring to an applied force. The force is applied at one end of a spring and the other side is fixed. The input to this model is the force applied in pounds. The two states are position and velocity at one end of the spring in feet and (feet/sec) respectively. The two outputs are: (1) spring position in feet, and (2) a direct feed-through term that measures the applied force. The state-space representation of this system is in file: Modif1.Qdr. The force in the original system input is defined in (lb) and it must be converted to (oz), the scaling factor $S=16$. The system output is in (feet) and it must be converted to (inches), the scaling factor $S=12$.

The diagram in Figure (3.1) shows both: the original and the scaled system in parallel. Both systems are equivalent. The original system is shown on the top. Its input is in (lb) and its two outputs are in (feet) and (lb). This system was transformed, as we shall see, to a new set of units, shown below. Its input is in (ounces) and its outputs are in (inches) and (ounces). The equivalence of the two systems is demonstrated by including unit conversion gains to rescale the inputs and the outputs of the modified system to match the original. The two systems are excited together in parallel and they respond identically, as expected.

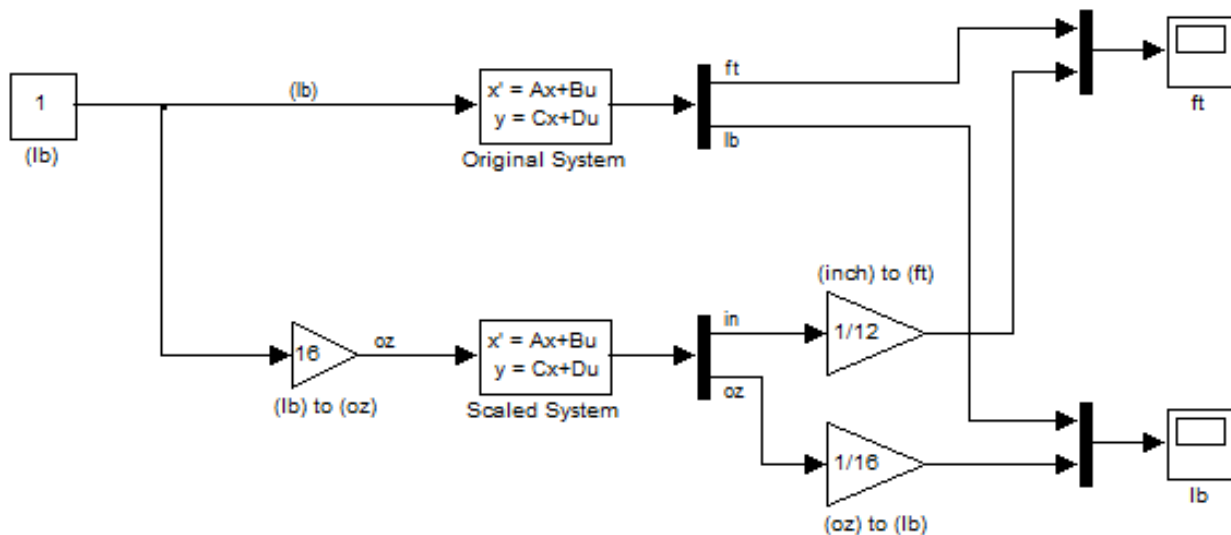
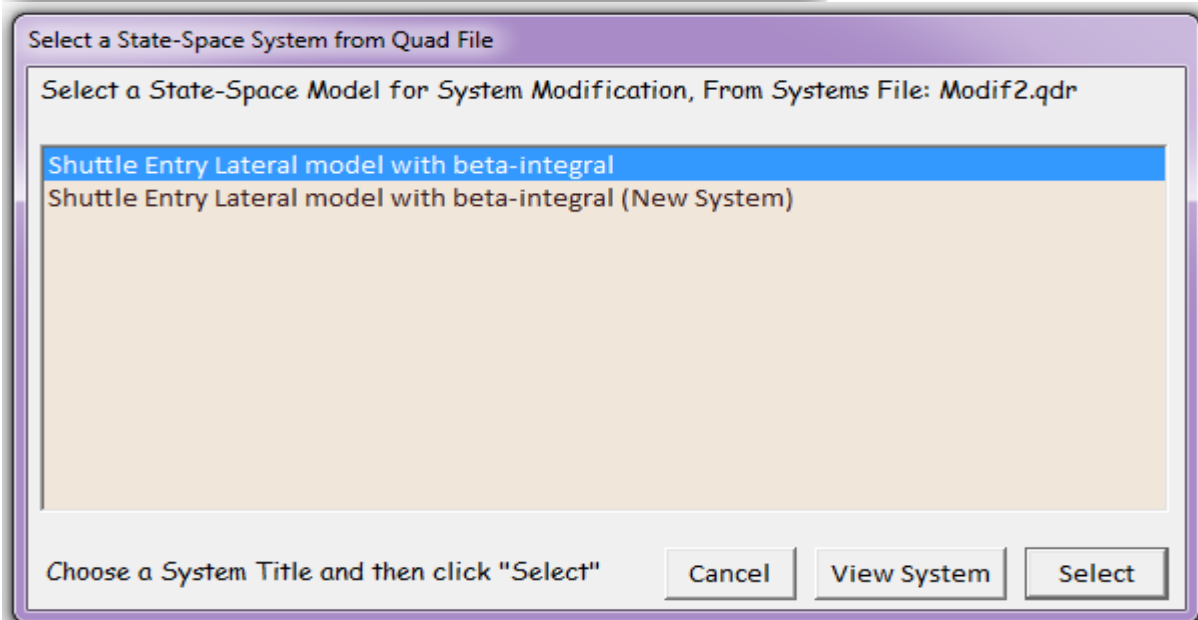
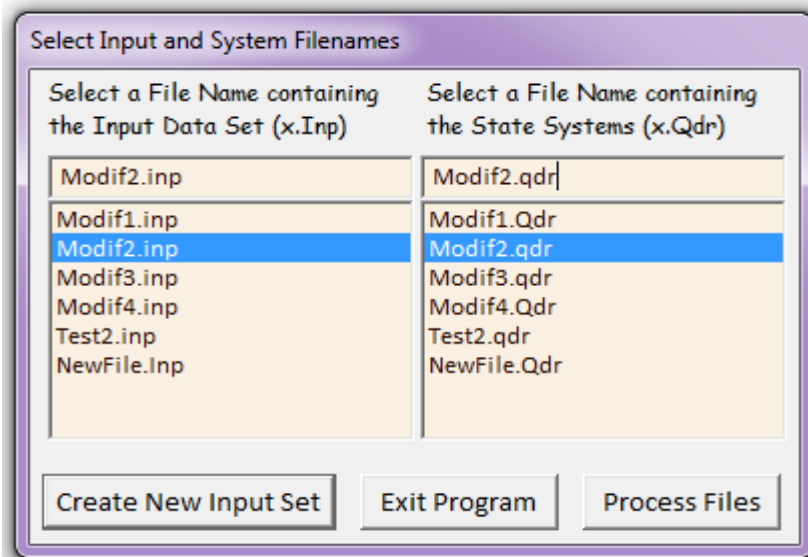


Figure (3.1) Comparison between the Original and the Scaled Systems

Re-scaling a system's input is performed by dividing with the scaling factor (S) the corresponding column in matrices B and D. Similarly, rescaling a system's output is performed by multiplying with the scaling factor (S) the corresponding row in matrices C and D. Rescaling the states affects only matrices A, B, and C. Notice, that it is not necessary to rescale the states in order to combine two

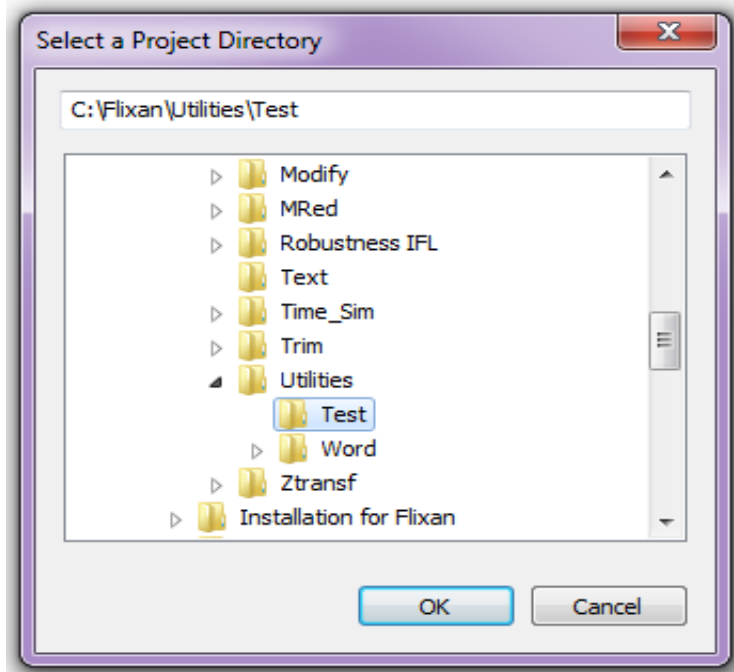
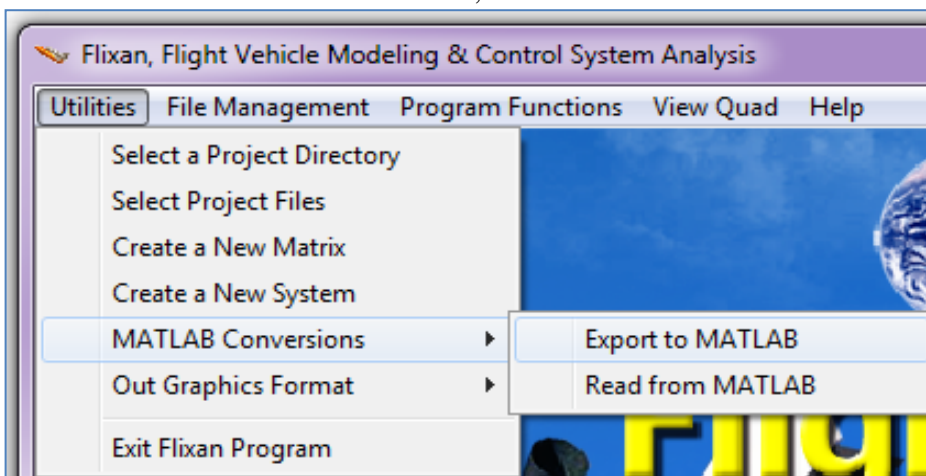
3.5 Creating a New Set of Outputs Interactively

Sometimes a system's output does not include sufficient variables. Some required states or derivatives do not appear in the output vector and the analyst would like to modify the output vector to include them. This option is used to replace a system's outputs with a new set of outputs from a combination of the original system variables. The new outputs can be selected from: (a) the original system outputs, (b) its output derivatives, (c) the old system states, and (d) the derivatives of the states. This utility is particularly useful in preparing synthesis models for control system design because design models often require additional output criteria to be optimized that may not necessarily appear in the original output. The original system to be modified in this example is "Shuttle Entry Lateral model with beta-integral" located in file "Modif2.Qdr". The output reconstruction data-set is not available yet so we will create it interactively and place it in file "Modif2.Inp". So we begin the program, as in previous examples, by selecting the two files from the filenames selection menu, and click on "Create a New Input Set" to begin the interactive data collection process. From the next menu select the original system from file "Modif2.Qdr", and click on "Select".



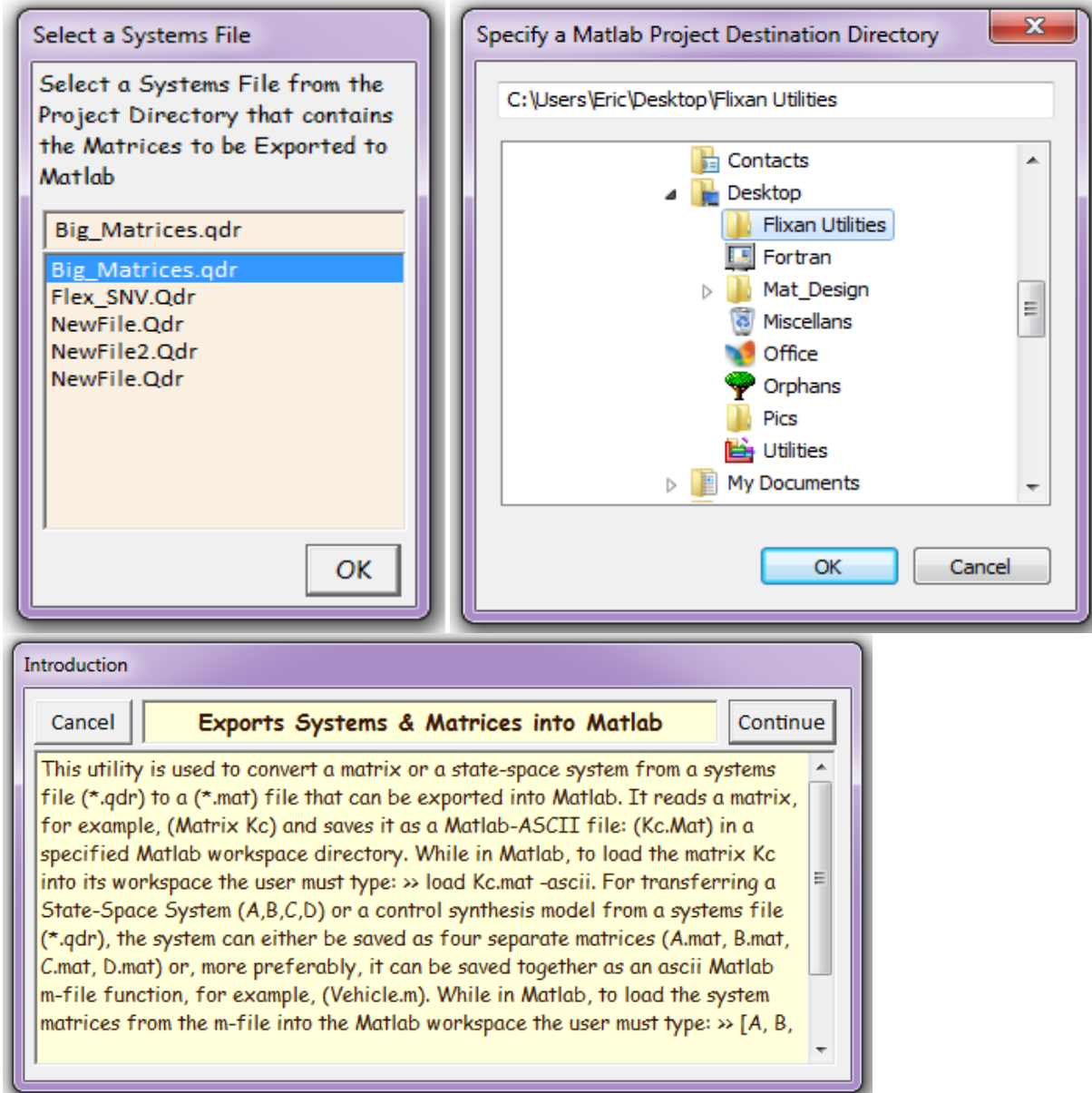
4.0 Utilities for Exporting data to Matlab and Importing data from Matlab

The Flixan program includes utility functions that convert Flixan data to Matlab format and also read data from Matlab generated files. The "Matlab exporting" utility reads matrices or state-space systems from systems files (.Qdr) and converts them into Matlab "ascii" matrices (.Mat) or m-file functions (.m), where they can be loaded into Matlab. The "from Matlab" importing utility reads Matlab "ascii" matrices (.Mat) and saves them in a systems file (.Qdr), either as a state-space system or as individual matrices using names other than [A,B,C,D], which identify a system. The Matlab data are not necessarily located in the same directory folder as the Flixan data. The user is prompted to specify separate directories, one for the data files used for the Flixan analysis, and a separate directory for the Matlab analysis containing the Simulink files (.Mdl), m-files (.m), and Matlab matrices (.Mat). The Flixan project directory is selected first. Then go to the Flixan main menu and click on "*Utilities*" to select the Matlab conversion utilities, as shown.



4.1 Exporting Data from Flixan to Matlab

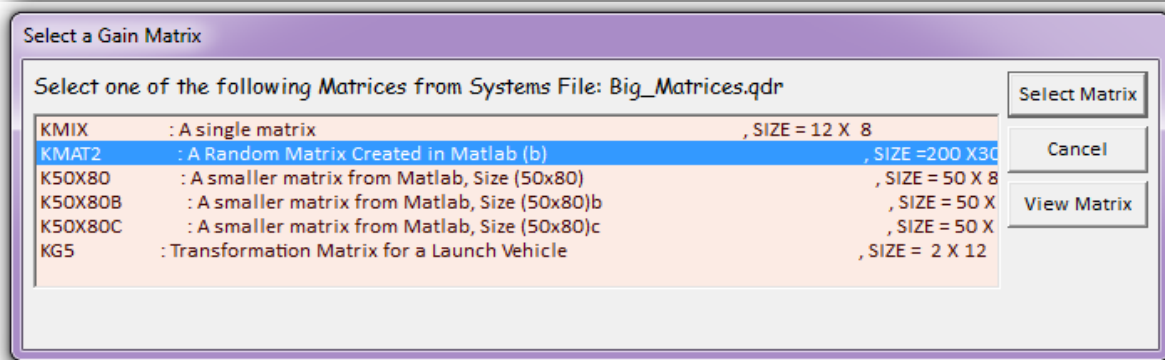
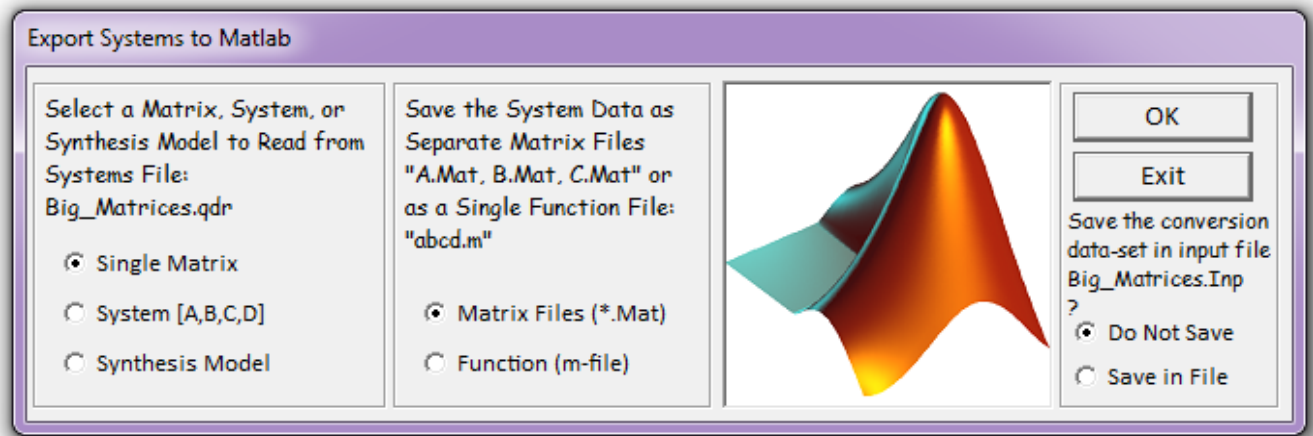
To convert systems or single matrices from Flixan files to Matlab format, go to "*Matlab Conversions*" and select "*Export to Matlab*". This will open a systems filename (.Qdr) selection menu where the user selects a systems file. This file contains the data to be exported to Matlab. The program also asks the user to select the Matlab destination directory which may be different from the location of the systems file.



A Matlab exporting dialog comes up from where the user has a choice to select between three different types of data to be transferred from the systems file to Matlab format: (a) a single gain matrix (other than A, B, C, D), (b) a state-space system [A, B, C, D], and (c) a synthesis model for H-infinity control design [A, B1, B2, C1, C2, D11, D12, D21, D22]. The user can also select the Matlab output format: either as "ascii" matrix files (.mat), or as function that includes several matrices in a single function m-file (.m). There is a third choice where the user may wish to save the "To Matlab" instructions data-set in the input file. This data-set can be used for reprocessing the input file in batch mode.

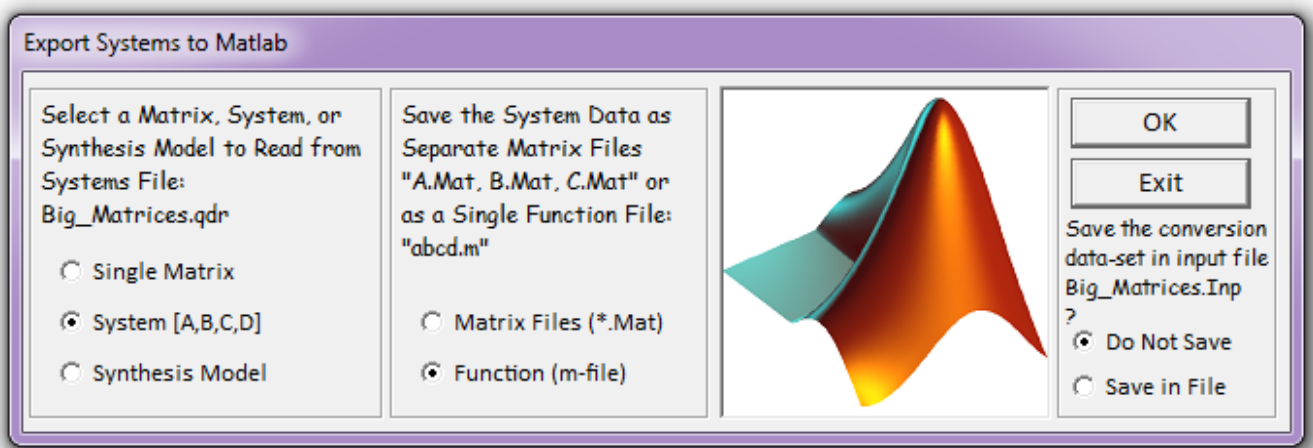
Exporting a Matrix

If the input data is a single matrix it will be converted into a Matlab (.mat) ascii file. The filename is the same as the matrix name in the (.Qdr) file. For example, if the matrix name was "Kmix2" in the systems file it will be saved as "Kmix2.mat" in the Matlab folder. The matrix to be transferred is selected from a menu that includes all the individual matrices, other than (A, B, C, D), which are saved in the selected systems file. The matrix name, a short description, and its size appear in the menu.



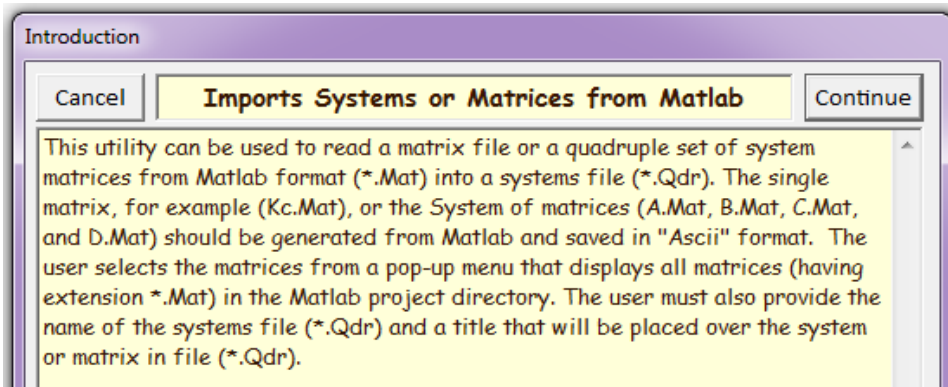
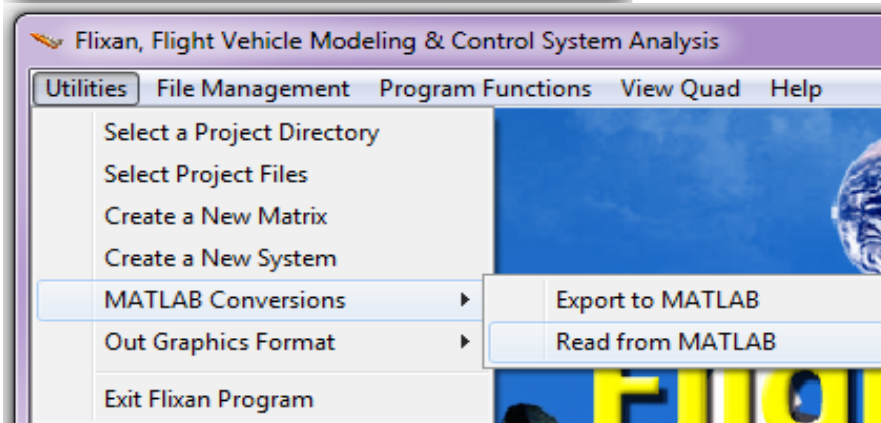
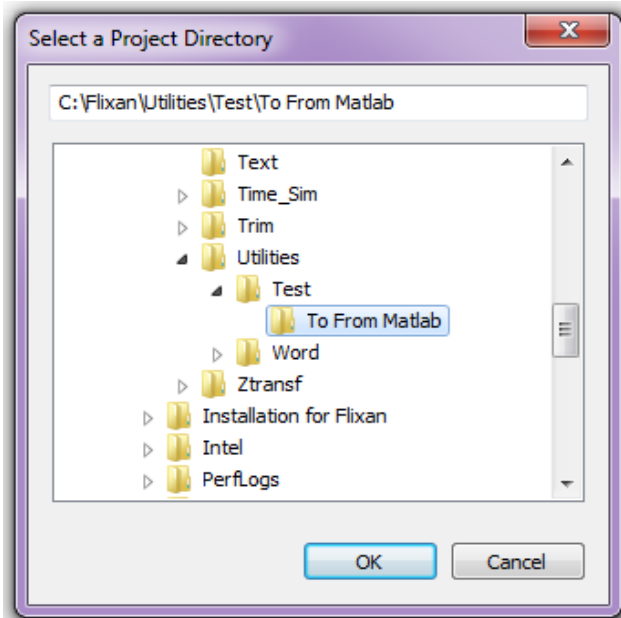
Exporting a System

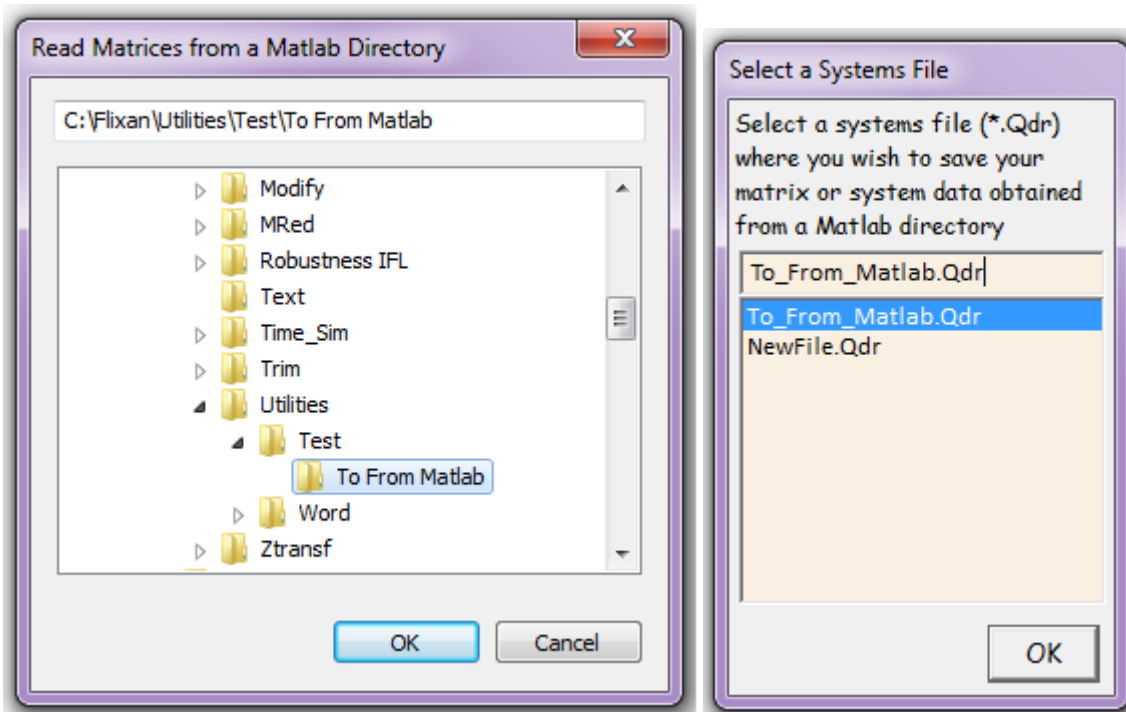
From the Matlab exporting dialog we have the choice to save the four state-space matrices either as individual matrices (A.mat, B.mat, C.mat, D.mat) or as a system function (m-file). In this example we select to save them as a function m-file, and click on "OK".



4.4 Transferring Matlab Generated Data to Flixan System Files

The process of reading Matlab data into Flixan system files (.Qdr) is similar. First, select the directory where the Flixan files are located and from Flixan main menu select "*Utilities*", "*Matlab Conversions*", and "*Read from Matlab*". You must also select the Matlab directory that contains the Matlab generated matrix files to be transferred. In this case shown both directories are the same. From the system filename selection menu select a systems file to save the imported matrices or systems.

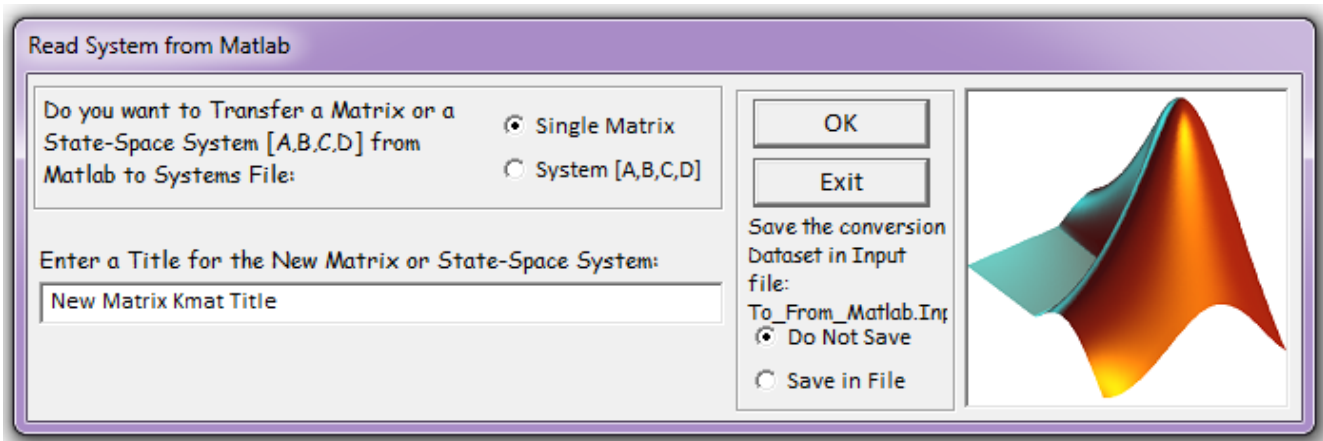




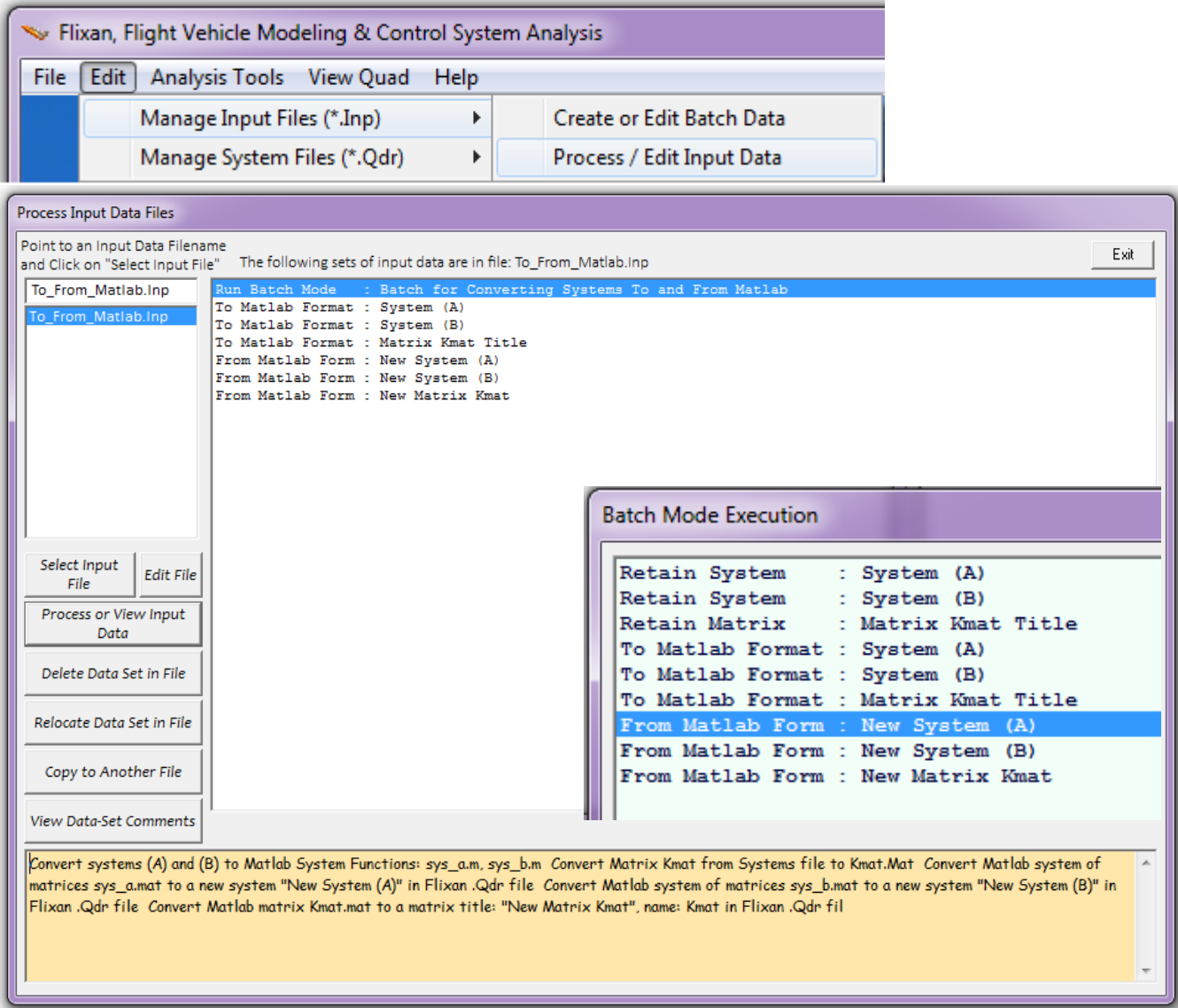
The following dialog is used to select if you want to read an individual matrix (other than A,B,C,D) or a state-space system. You must also enter a title for this system or matrix different from other elements in the systems file, and click OK. The user may also save the "From Matlab" instructions data-set in the input file. This data-set can be used for reprocessing the input file in batch mode at a later time.

Reading a Matrix from Matlab

When the input is a matrix the user clicks on "Single Matrix", enters its new title, and clicks on "OK". The Matlab generated matrix is expected to be a (.Mat) file in ascii format. The following menu shows all (.Mat) files located the selected Matlab folder. The user selects one of the files "Kmat.mat" in this case, which is a single matrix, and clicks on "Select File". The other two (.Mat) files are systems containing quadruple matrices. The matrix will then be saved in the selected systems file as "Matrix Kmat", under the title entered: "New Matrix Kmat Title".



This input file can now be processed in batch mode by running the batch set located at the top of the file. From the Flixan main menu select "Edit", then "Manage Input Files" and then "Process/ Edit Input Data". From the next dialog use the left menu to select the input file "To_From_Matlab.Inp" and click on "Select Input File". The menu on the right shows the titles of the data-sets located in this file, also the program that will process each set. Select the first title which is the batch set and click on "Process or View Input Data", and the batch set will process the data. The Matlab generated systems and matrix are saved in file *To_From_Matlab.Qdr*". Notice that the original systems and matrix are retained in the systems file after reprocessing. A display pop-up highlighting the batch statements as they are executing.



5.0 Continuous to Discrete System Transformations

This program is mainly used for transforming continuous state-space systems to discrete systems described by a set of state difference equations.

$$x(k + 1) = Fx(k) + Gu(k);$$

$$y(k) = Hx(k) + Eu(k)$$

It is also used for transforming a discrete system from the z-plane to the w-plane, and from the w-plane back to the z-plane. The program reads the state-space system data of the system to be converted from a systems file ".Qdr", the transformation instructions from an input file ".Inp", and can perform the following transforms:

- Transform a continuous system to a discrete system using the Z-transform.
- Transform a continuous system to a discrete system using the Tustin transform.
- Transform a continuous system with delays at its inputs to a discrete system using the Z-transform. The delays at the system inputs are (from 0 to δT), where δT is the sampling period.
- Transform a discrete system defined in the z-plane to a discrete system in the w-plane, and
- Transform a discrete system in the defined w-plane to a discrete system in the z-plane.

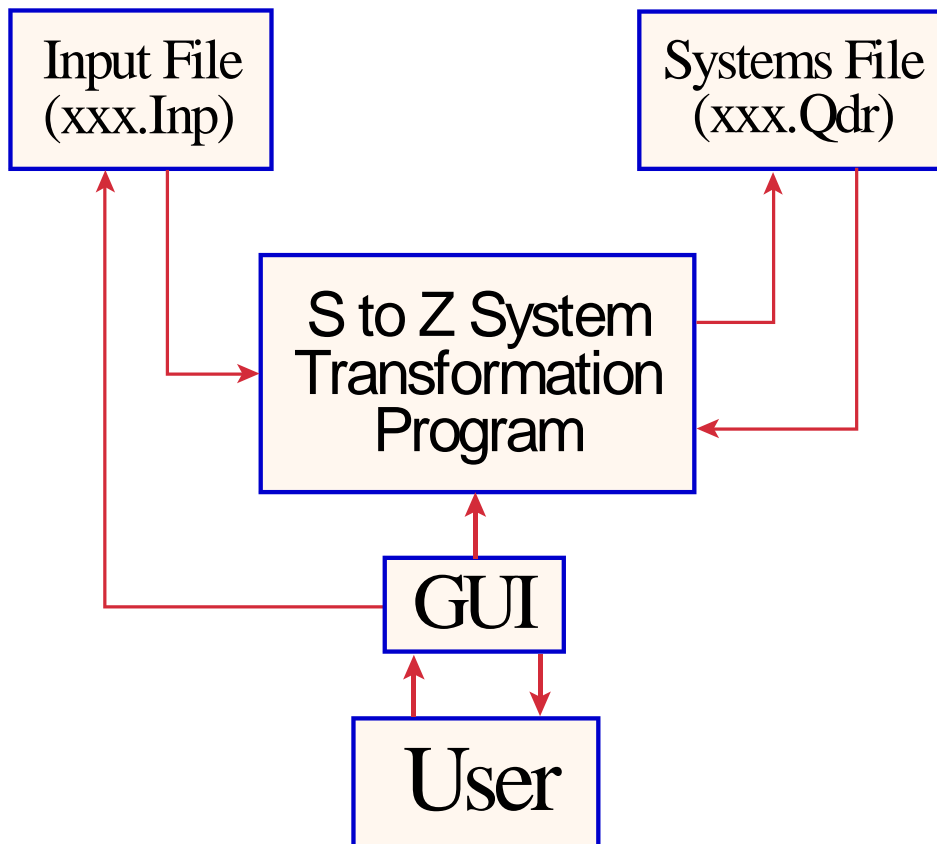


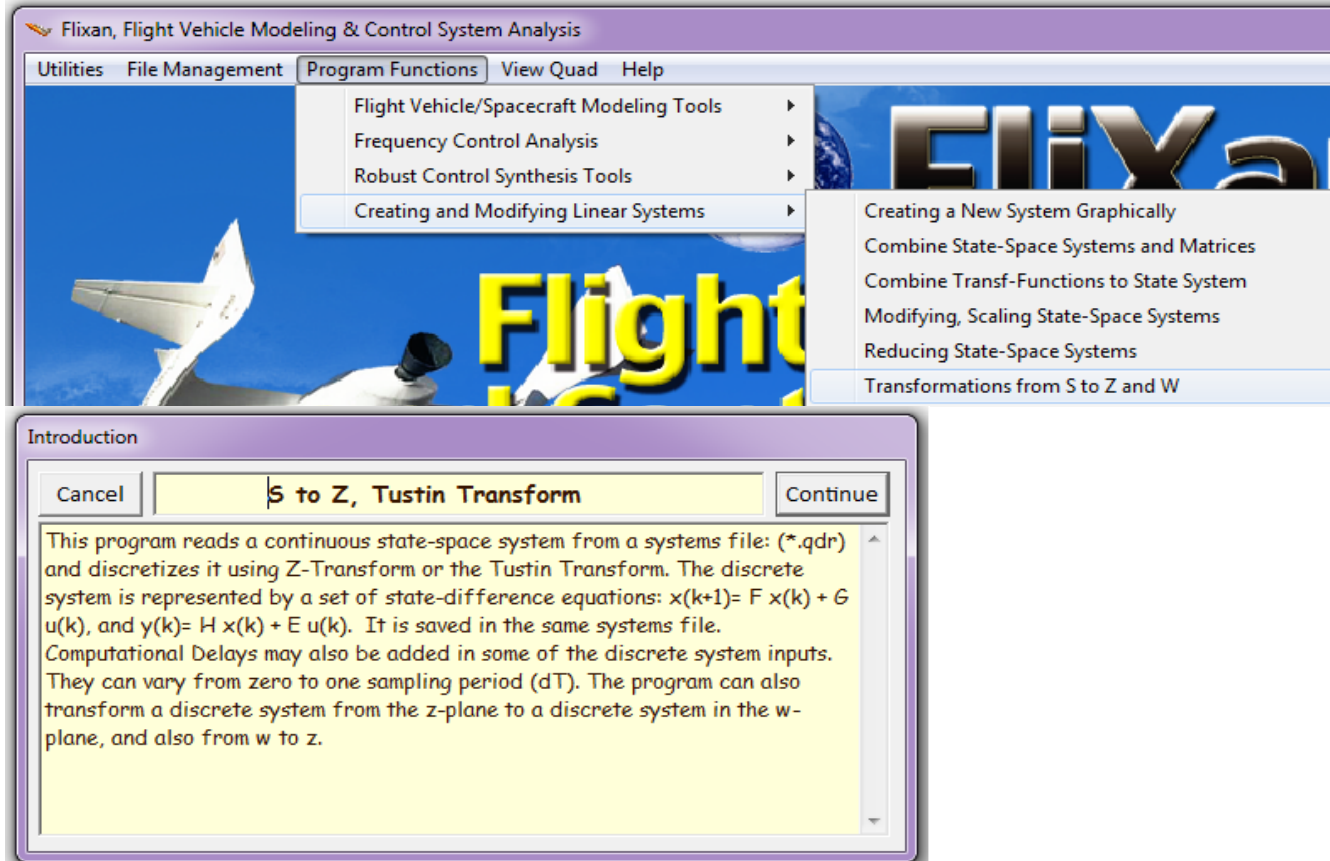
Figure (5.1) Systems S to Z to W Transformation Program

5.1 Running the Z-Transform Program

The Z-Transformation program like most Flixan programs requires two files: an input data file ".Inp" and a systems file ".Qdr". The systems file initially contains the state-space system to be transformed by the program. The transformed system will also be saved in this file by the z-transformation program. The input file contains a set of instructions defining the process, such as the one shown below. The data set first contains a line that identifies the data process: "TRANSFORM A SYSTEM (S-Z-W)". The next line is the title of the transformed new system (blue). Some comment lines (shown in green) may be included below the title. Next line is the title of the original system to be transformed (blue). The last line (black) shows the type of transformation to be performed and the sampling period (δT) of the transformed system, for example:

```
TRANSFORM A SYSTEM (S-Z-W) ... (Titles of the New and Old Systems)
Flexible System (7-Modes) (Z-Transform)
! This data-set discretizes the system "Flexible System (7-Modes)"
! The new discrete system title is "Flexible System (7-Modes) (Z-Transform)"
Flexible System (7-Modes)
From S-plane to Z-plane using the Z-Transform, dT= 0.002
```

To run the z-transform program start from the Flixan main menu, click on “*Program Functions*”, select “*Creating and Modifying Linear Systems*”, and then click on “*Transformations from S to Z and W*”. Then use the following filename selection dialog to select the input data and the systems files from the menus, as shown below, and click on “*Process Files*”. The input file contains the system transformation instructions previously created such as the ones above. If not, they must be defined them interactively by clicking on “*Create New Input Set*”.



5.4 Types of Transformations

The program basically uses two methods to discretize a continuous system. The z-transform and the Tustin transformation methods. A third option that adds delays at the system inputs is also included.

S to Z via the Z-Transform

A continuous state-space system (A, B, C, D) can be transformed to a discrete system described by a set of state difference matrices (F, G, H, E) using the following z-transformation.

$$F = e^{At}; \quad G = (e^{At} - I)A^{-1}B; \quad H = C; \quad E = D$$

The program uses the following equation to calculate the matrix exponential $\{ e^{AT} \}$.

$$e^{At} = (I + e^{-At})^{-1} (I + e^{At})$$

$$e^{At} = \left(I - \frac{AT}{2} + \frac{(AT)^2}{4} - \frac{(AT)^3}{12} \right)^{-1} \left(I + \frac{AT}{2} + \frac{(AT)^2}{4} + \frac{(AT)^3}{12} \right)$$

S to Z via the Tustin-Transform

The second method used for transforming a continuous state-space system (A, B, C, D) to a discrete system described in terms of state difference matrices (F, G, H, E) is the Tustin transformation described by the following equations.

$$F = F_1^{-1} F_2; \quad G = 2F_1^{-2} \left(\frac{BT}{2} \right)$$

$$H = C; \quad E = D + CF_1^{-1} \left(\frac{BT}{2} \right) \text{ where}$$

$$F_1 = \left(I - \frac{AT}{2} \right) \quad F_2 = \left(I + \frac{AT}{2} \right)$$

Z-Transform of a System with an Input Delay

This option is used for including additional delays (δ) sec at the inputs of a system, where the delays are less than one sampling period ($\delta < T$). It requires that the system be z-transformed at two different rates. Figure (5-2) is a timing diagram showing an input/output sequence of a discrete system with a delay (δ). The system's output and state equations are shown below. The state vector propagation is described in two steps:

$$y(n) = Cx(n) + Du(n-1)$$

$$x(n+\delta) = \Phi(\delta)x(n) + \Gamma(\delta)u(n-1)$$

$$x(n+1) = \Phi(T-\delta)x(n+\delta) + \Gamma(T-\delta)u(n)$$

The matrices $\Phi(\cdot)$ and $\Gamma(\cdot)$ are the discrete system's state transition matrix and input matrix. This approach is implemented by z-transforming the continuous system at two different sampling rates: at (δ) , and at $(T-\delta)$ seconds. By combining the state difference equations and augmenting the state vector, the discrete system with the computational delay can be expressed by the following augmented state-space representation.

$$\begin{bmatrix} x(n+1) \\ u(n) \end{bmatrix} = \begin{bmatrix} \Phi(T) & \Gamma(T) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(n) \\ u(n-1) \end{bmatrix} + \begin{bmatrix} \Gamma(T-d) \\ I \end{bmatrix} u(n)$$

$$y(n) = [C \quad D] \begin{bmatrix} x(n) \\ u(n-1) \end{bmatrix} \text{ where}$$

$$\Phi(T) = \Phi(T-\delta)\Phi(d) \text{ and } \Gamma(T) = \Phi(T-\delta)\Gamma(d)$$

When the z-transform option with delay is selected the program opens the following dialog and asks for the transport delay (δ) , in addition to the sampling period (T) . The delay (δ) must be less than (T) . Note, that for delays which are multiple to one sampling period (nT) they can be implemented using discrete transfer functions $\{ z^{-n} \}$ in series with the z-transformed system, see example 1 below.

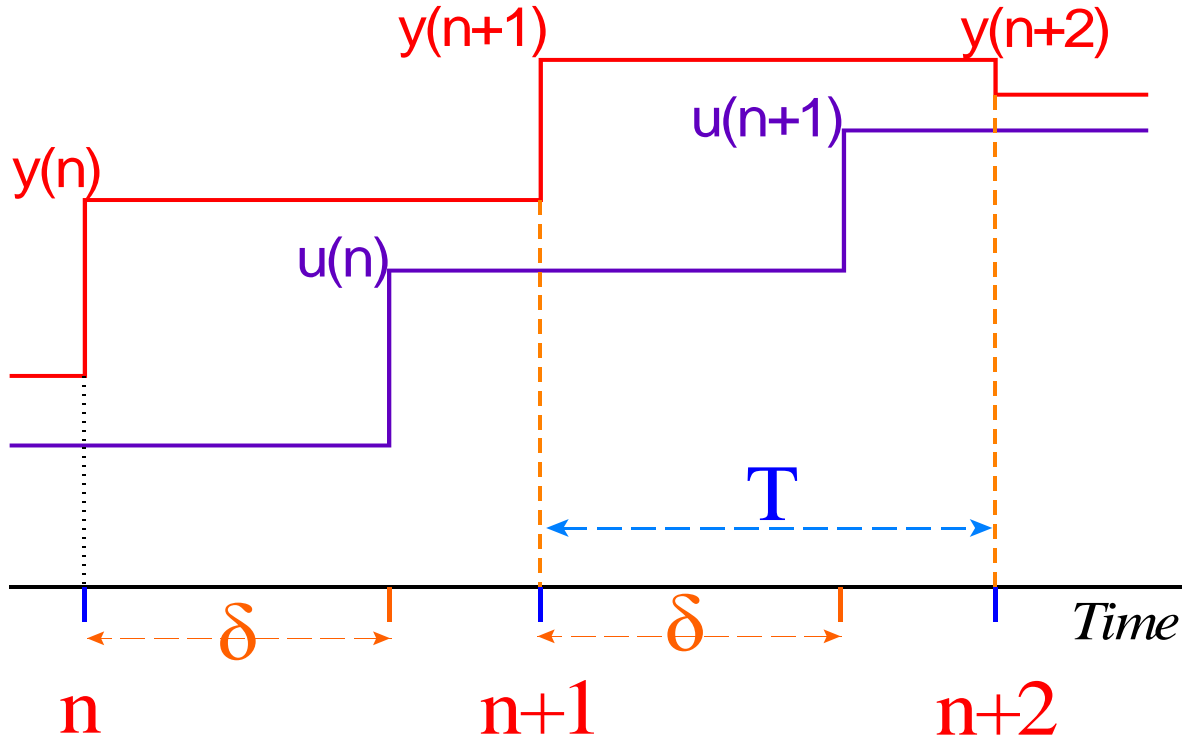
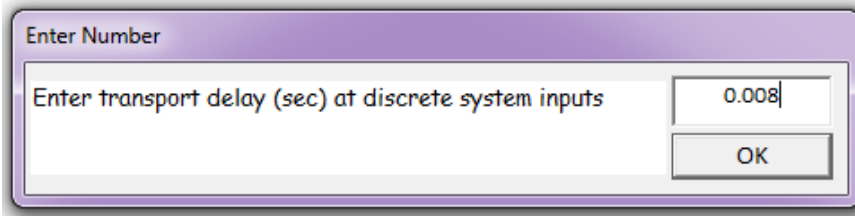


Figure (5.2) Input/ Output Timing Diagram

5.5 More Z-Transform Examples

The following examples demonstrate some more applications of the Z-transform program. The first example uses a model of a lightly damped structure. The second example is a lateral model of a Lockheed 1011 aircraft. The third example is a model of the Shuttle vehicle during ascent which includes structural flexibility and fuel sloshing.

Flexible Plate

The first example is a SISO model of a flexible plate consisting of seven flex resonances. The input to this model is force at a certain location perpendicular to the surface and the output is acceleration at a different location. This system is in file: "Ztran_Ex1.Qdr", located in folder "C:\Flixan\Ztransf\Examples\Ex-1", and its title is: "Flexible System (7-Modes)". The transformations to be performed are already defined in the input file "Ztran_Ex1.Inp", shown below. The z-transform program will be used to discretize this system using the three options: (a) Z-Transform, (b) Tustin Transform, and (c) Z-Transform with delay. An additional system is created by combining the z-transformed system with one sample delay in series. We will then analyze the three discrete systems by comparing their frequency responses against those of the continuous system. Similar transformations are performed for TVC engine actuator model located in the same files. The input file for this example is shown below and it will be processed in batch mode. The transformed new systems are also converted into Matlab format for further analysis.

```
BATCH MODE INSTRUCTIONS .....
Batch for discretizing two continuous systems with various delays
! This batch discretizes a continuous system with flex modes and introduces a delay
! of one sampling period at its input. It discretizes also the system using the Tustin
! transform, and also using the z-transform with a delay of 3/4 (dT). The discretized
! systems are then compared using frequency response analysis.
!
! This batch also calculates three types of discretizations for an actuator model using a
! sampling period of 2 msec.
!
Retain System      : Flexible System (7-Modes)
Retain System      : TVC Engine Actuator Example ...
!
!                FLEXIBLE PLATE EXAMPLE
!
Transf-Function    : One Sample Delay, z^{-1}, dT=0.0002
S-Z-Transform      : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
S-Z-Transform      : Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
S-Z-Transform      : Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
System Connection : Z-Transformed Flexible System with One Sample Delay at the Input
To Matlab Format   : Flexible System (7-Modes)
To Matlab Format   : Z-Transformed Flexible System with One Sample Delay at the Input
To Matlab Format   : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
To Matlab Format   : Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
To Matlab Format   : Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
!
!                ACTUATOR EXAMPLE
!
S-Z-Transform      : TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
S-Z-Transform      : TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
S-Z-Transform      : TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
To Matlab Format   : TVC Engine Actuator Example ...
To Matlab Format   : TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
To Matlab Format   : TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
To Matlab Format   : TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
```

 FLEX PLATE EXAMPLE

SYSTEM OF TRANSFER FUNCTIONS ...

One Sample Delay, z^{-1} , $dT=0.0002$

! Delay implemented as discrete transfer function

Discrete, $dt= 0.20000E-03$

TF. Block # 1 0.0002 sec Delay

Order of Numer, Denom= 1 1

Numer 0.00000 1.00000

Denom 1.00000 0.00000

 Block #, from Input #, Gain:

1 1 1.00000

.....

Outpt #, from Block #, Gain:

1 1 1.00000

.....

 INTERCONNECTION OF SYSTEMS

Z-Transformed Flexible System with One Sample Delay at the Input

!

! This system is obtained by combining the "(Z-Transform)" system in

! series with the "One Sample Delay $\{z^{-1}\}$ " system.

!

Titles of the Systems to be Connected ...

Title 1 One Sample Delay, z^{-1} , $dT=0.0002$

Title 2 Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)

.....

SYSTEM INPUTS TO SUBSYSTEM 1

System Input 1 to Subsystem 1, Input 1, Gain= 1.00000

.....

SYSTEM OUTPUTS FROM SUBSYSTEM 2

System Output 1 from Subsystem 2, Output 1, Gain= 1.00000

.....

SUBSYSTEM NO 1 GOES TO SUBSYSTEM NO 2

Subsystem 1, Output 1 to Subsystem 2, Input 1, Gain= 1.00000

.....

Definitions of Inputs = 1

One Sample Delay

Definitions of Outputs = 1

Flexible System (7-Modes)

 TRANSFORM A SYSTEM (S-Z-W) (New system title, Comments, Old System title, Transform)

Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)

! Continuous System Model: Acceleration of a Flexible Plate at a point in response to a Force

! Input which is applied at a different point. Z-Transformed at 0.0002 sec

Flexible System (7-Modes)

From S-plane to Z-plane using the Z-Transform, $dT= 0.20000E-03$

 TRANSFORM A SYSTEM (S-Z-W) (New system title, Comments, Old System title, Transform)

Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)

! Continuous System Model: Acceleration of a Flexible Plate at a point in response to a Force

! Input which is applied at a different point. Discretized using Tustin Transform, $dT=0.0002$ sec

Flexible System (7-Modes)

From S-plane to Z-plane using the Tustin-Transform, $dT= 0.20000E-03$

 TRANSFORM A SYSTEM (S-Z-W) (New system title, Comments, Old System title, Transform)

Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)

! Continuous System Model: Acceleration of a Flexible Plate at a point in response to a Force

! input which is applied at a different point. Z-Transformed at 0.0002 sec with 0.00015 sec Delay

Flexible System (7-Modes)

From S-plane to Z-plane using the Z-Transform with Delay, dT , Delay= $0.20000E-03$ $0.15000E-03$

 CONVERT TO MATLAB FORMAT (Title, System/Matrix, m-filename)

Flexible System (7-Modes)

System

Flex7_s.m

 CONVERT TO MATLAB FORMAT (Title, System/Matrix, m-filename)

Z-Transformed Flexible System with One Sample Delay at the Input

System

Flex7_z1.m

```

CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
System
Flex7_z.m
-----
CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
System
Flex7_zt.m
-----
CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
System
Flex7_zd.m
-----
ACTUATOR EXAMPLE
-----
TRANSFORM A SYSTEM (S-Z-W) ..... (New system title, Comments, Old System title, Transform)
TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
! The following system is an actuator model for the Space Shuttle main engine operating during
! nominal conditions. It must be discretized with a sample period of 0.002 seconds
TVC Engine Actuator Example ...
From S-plane to Z-plane using the Z-Transform, dT= 0.002
-----
TRANSFORM A SYSTEM (S-Z-W) ..... (New system title, Comments, Old System title, Transform)
TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
! The following system is an actuator model for the Space Shuttle main engine operating during
! nominal conditions. It must be discretized with a sample period of 0.002 seconds
TVC Engine Actuator Example ...
From S-plane to Z-plane using the Tustin-Transform, dT= 0.002
-----
TRANSFORM A SYSTEM (S-Z-W) ..... (New system title, Comments, Old System title, Transform)
TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
! The following system is an actuator model for the Space Shuttle main engine operating during
! nominal conditions. It must be discretized with a sample period of 0.002 seconds
TVC Engine Actuator Example ...
From S-plane to Z-plane using the Z-Transform with Delay, dT, Delay= 0.20000E-02 0.15000E-02
-----
CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
TVC Engine Actuator Example ...
System
Actuat_s.m
-----
CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
System
Actuat_z.m
-----
CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
System
Actuat_zt.m
-----
CONVERT TO MATLAB FORMAT ..... (Title, System/Matrix, m-filename)
TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
System
Actuat_zd.m
-----

```

The above input file is now processed in batch mode by running the batch set located on the top of the file. From the Flixan main menu select "Edit", then "Manage Input Files" and then "Process/ Edit Input Data". From the following dialog use the left menu to select the input file "Ztran_Ex1.Inp" and click on "Select Input File". The menu on the right shows the titles of the data-sets located in this file, also the program that will process each set. Select the first title which is the batch set, click on "Process or View Input Data", and the batch set will process the data, and save the transformed systems in file "Ztran_Ex1.Qdr". Notice that the two original continuous systems are retained in the systems file after reprocessing. A display shows the batch statements as they are executing. A Matlab script "run_flx.m", located in the same folder, loads the systems and calculates their frequency responses, as shown below.

Process Input Data Files

Point to an Input Data Filename and Click on "Select Input File" The following sets of input data are in file: Ztran_Ex1.inp

Exit

- Ztran_Ex1.inp
- Ztran_Ex1.inp

```
Run Batch Mode : Batch for discretizing two continuous systems with various delays
Transf-Functions : One Sample Delay, z^{-1}, dT=0.0002
System Connection: Z-Transformed Flexible System with One Sample Delay at the Input
S-Z-Transform : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
S-Z-Transform : Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
S-Z-Transform : Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
To Matlab Format : Flexible System (7-Modes)
To Matlab Format : Z-Transformed Flexible System with One Sample Delay at the Input
To Matlab Format : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
To Matlab Format : Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
To Matlab Format : Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
S-Z-Transform : TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
S-Z-Transform : TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
S-Z-Transform : TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
To Matlab Format : TVC Engine Actuator Example ...
To Matlab Format : TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
To Matlab Format : TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
To Matlab Format : TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
```

- Select Input File
- Edit File
- Process or View Input Data
- Delete Data Set in File
- Relocate Data Set in File
- Copy to Another File
- View Data-Set Comments

Replace Systems File?

The systems filename: Ztran_Ex1.Qdr already exists. Do you want to create it again?

Yes No

This batch discretizes a continuous system with flex modes and introduces a delay of one sampling period at its input. It discretizes also the system using the Tustin transform, and also using the z-transform with a delay of 3/4 (dT). The discretized systems are then compared using frequency response analysis. This batch also calculates three types of discretizations for an actuator model using a sampling period of 2 msec.

Process Input Data Files

Point to an Input Data Filename and Click on "Select Input File" The following sets of input data are in file: Ztran_Ex1.inp

Exit

- Ztran_Ex1.inp
- Ztran_Ex1.inp

```
Run Batch Mode : Batch for discretizing two continuous systems with various delays
Transf-Functions : One Sample Delay, z^{-1}, dT=0.0002
System Connection: Z-Transformed Flexible System with One Sample Delay at the Input
S-Z-Transform : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
```

Batch Mode Execution

```
Retain System : Flexible System (7-Modes)
Retain System : TVC Engine Actuator Example ...
Transf-Function : One Sample Delay, z^{-1}, dT=0.0002
S-Z-Transform : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
S-Z-Transform : Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
S-Z-Transform : Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
System Connection: Z-Transformed Flexible System with One Sample Delay at the Input
To Matlab Format : Flexible System (7-Modes)
To Matlab Format : Z-Transformed Flexible System with One Sample Delay at the Input
To Matlab Format : Flexible System (7-Modes) (Z-Transformed at 0.0002 sec)
To Matlab Format : Flexible System (7-Modes) (Tustin-Transformed at 0.0002 sec)
To Matlab Format : Flexible System (7-Modes) (Z-Transformed with 0.00015 sec Delay)
S-Z-Transform : TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
S-Z-Transform : TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
S-Z-Transform : TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
To Matlab Format : TVC Engine Actuator Example ...
To Matlab Format : TVC Engine Actuator Example ... (Z-Transformed at 2 msec)
To Matlab Format : TVC Engine Actuator Example ... (Tustin-Transformed at 2 msec)
To Matlab Format : TVC Engine Actuator Example ... (Z-Transformed at 2 msec with 1.5 msec delay)
```

View Data-Set Comments

This batch discretizes a continuous system with flex modes and introduces a delay of one sampling period at its input. It discretizes also the system using the Tustin transform, and also using the z-transform with a delay of 3/4 (dT). The discretized systems are then compared using frequency response analysis. This batch also calculates three types of discretizations for an actuator model using a sampling period of 2 msec.

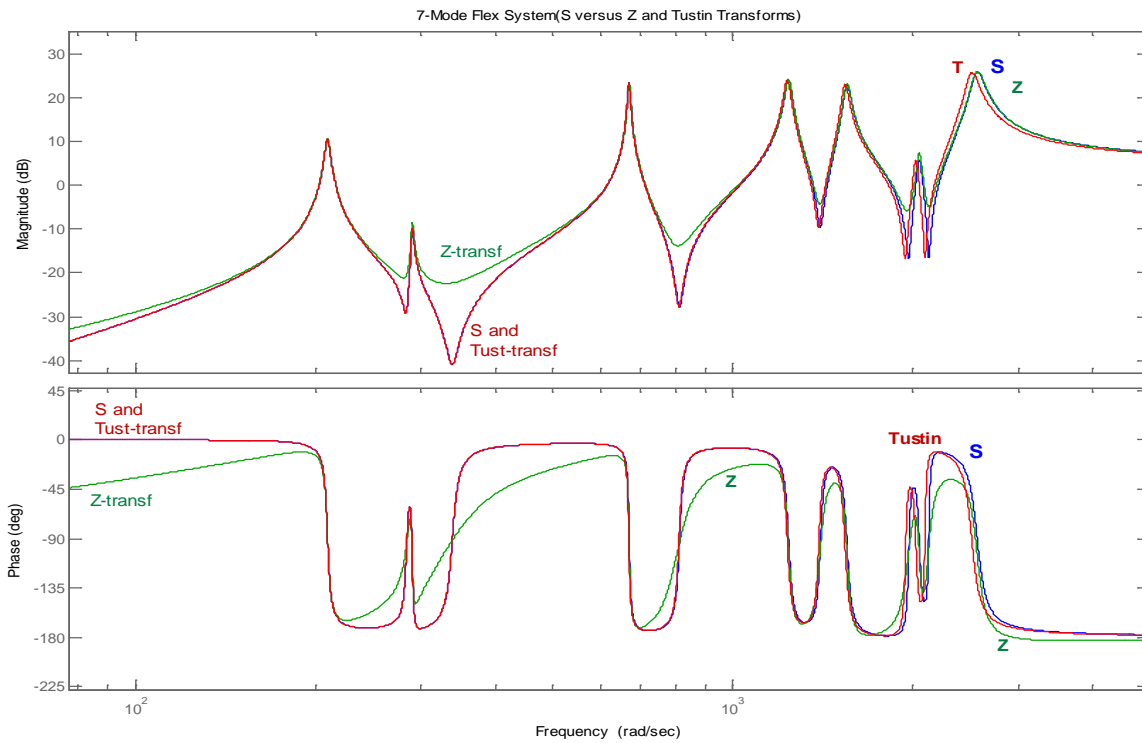


Figure (5.3) Frequency Response comparison between the continuous, the z-transformed, and the Tustin-transformed systems. The Tustin transform is a closer match to the continuous system. It causes less phase lag but it creates a slight shift at the high frequency resonance.

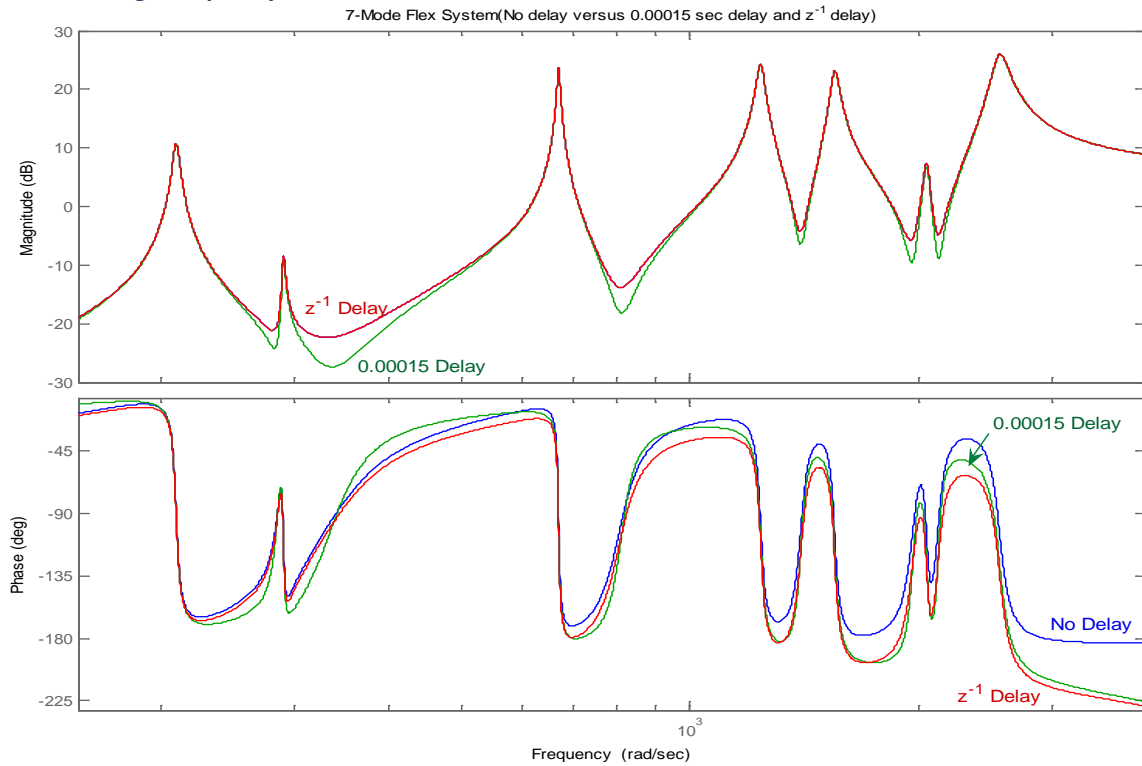


Figure (5.4) Frequency Response comparison between three z-transformed systems, (a) the z-transformed system without delays, (b) the system with the 0.00015 sec input delay, and (c) the system with the (z^{-1}) delay at its input. Notice that the z-transformed systems with the delays have significantly more phase lag at high frequencies than the z-transformed system without delay, and the system with the one sampling period (T) delay has more phase lag than the system with (0.75T) delay, as expected.

6.0 Managing Input/Output Files

The Flixan program includes several utility programs for managing the input data and systems files. The input files (.Inp) contain sets of data or instructions which are processed by various Flixan programs generating matrices and state-space systems in (.Qdr) files for system analysis. Input data files typically contain multiple sets of data and system files may also contain several systems and single matrices related to a particular vehicle or project. All files associated with the same project are located in the same directory which is typically named after the vehicle name or flight condition. The purpose of the file management utilities is to organize the files and process the input data. They perform functions, such as: cleaning the files from unused data, viewing and editing user comments, reordering the sequence of some functions, editing the files, and for creating and editing batch sets for fast processing. The file management utilities are user friendly and fully interactive by means of window menus and dialogs.

6.1 Input Data files (*.Inp)

The input data files are typically created interactively by the various Flixan utilities and programs. They can also be created by the user, by copying, pasting and modifying previously generated data sets using a text editor. Each input file contains multiple sets of input data that correspond to various Flixan programs, such as: flight vehicle or actuator modeling, system or transfer function interconnections, system discretization or restructuring, effector mixing logic, etc. Each program processes its corresponding data and creates a system or a matrix. The data sets in an input file can either be processed interactively by selecting the corresponding utility through the Flixan menus or they can be processed through the file manager utility. They can also be processed much faster by running a batch data-set, again through the file manager utility. The first line of a data-set (red) identifies the type of data and the utility that will process it. For example, FLIGHT VEHICLE INPUT DATA, INTERCONNECTION OF SYSTEMS, ACTUATOR INPUT DATA, MIXING LOGIC, BATCH MODE INSTRUCTIONS, etc. The second line is a unique title for the data-set (blue) and it should be less than 100 characters long. The same title is also used to identify the system or the matrix that will be created by the program or utility and it is saved in a systems file. The lines below the title, starting with an exclamation mark (!) are user comments (green) which are inserted there, either directly by editing the input data file, or transferred there by another utility. Each comment line should be less than 100 characters long and they should be limited to 10 lines. A sample input data file is shown below.

```
BATCH MODE INSTRUCTIONS .....
Batch for calculating the Closed-Loop model for a Zulu Rocket
! This batch set creates a state-space model for a rigid Zulu rocket during first stage
! at Max-Q and then extracts the longitudinal subsystem. It creates also a flex vehicle
! model with fuel sloshing and extracts also the longitudinal flex subsystem.
! It creates also state-space systems for the angle of attack estimator, the flight
! control system, and the closed-loop system. The state-space systems are converted to
! Matlab format.
Flight Vehicle      : Zulu, Stage-1, Max-Q, Rigid Body
System Modificat   : Zulu, Stage-1, Max-Q, Rigid Body, Pitch Axis
Flight Vehicle     : Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh
System Modificat   : Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh, Pitch Axis
Transf-Function    : Alpha Estimator
Transf-Function    : Pitch Flight Control System
System Connection  : Pitch Closed-Loop System
To Matlab Format   : Zulu, Stage-1, Max-Q, Rigid Body, Pitch Axis
-----
```

FLIGHT VEHICLE INPUT DATA

Zulu, Stage-1, Max-Q, Rigid Body

! This is a Launch Vehicle during First Stage at Maximum Dynamic Pressure. It is launched from
! a high altitude to maximize orbit performance. At Max-Q and while flying at 2.5 Mach it
! experiences a strong wind shear that causes 4 deg of angle of attack. The vehicle has two side
! liquid oxygen rockets that can provide 320,000 (lb) of thrust each. The two fuel tanks, one
! on each side cause a significant amount of slosh disturbance.

! Body Axes Output,Attitude=Euler Angles,Without GAFD, No Turn Coordination

Vehicle Mass (lb-sec ² /ft), Gravity Accelerat. (g) (ft/sec ²), Earth Radius (Re) (ft)	:	8878.0	32.174	0.20896E+08	
Moments and products of Inertias Ixx, Iyy, Izz, Ixy, Ixz, Iyz, in (lb-sec ² -ft)	:	519500.0	1671000.0	2110000.0	0.0000
CG location with respect to the Vehicle Reference Point, Xcg, Ycg, Zcg, in (feet)	:	-126.05	0.0000	0.007	
Vehicle Mach Number, Velocity Vo (ft/sec), Dynamic Pressure (psf), Altitude (feet)	:	2.5	2000.0	220.00	83000.
Inertial Acceleration Vo_dot, Sensed Body Axes Accelerations Ax,Ay,Az (ft/sec ²)	:	62.000	54.000	0.0000	21.000
Angles of Attack and Sideslip (deg), alpha, beta rates (deg/sec)	:	4.0000	0.0000	0.0000	0.0000
Vehicle Attitude Euler Angles, Phi_o,Thet_o,Psi_o (deg), Body Rates Po,Qo,Ro (deg/sec)	:	0.0000	44.000	0.0000	0.0000
Wind Gust Vel wrt Vehi (Azim & Elev) angles (deg), or Force(lb), Torque(ft-lb), locat:xyz:			Gust	45.0	45.0
Surface Reference Area (feet ²), Mean Aerodynamic Chord (ft), Wing Span in (feet)	:	47.3	7.7000	7.7000	
Aero Moment Reference Center (Xmrc,Ymrc,Zmrc) Location in (ft), {Partial_rho/ Partial_H}	:	-127.2	0.0000	0.0000	0.0000
Aero Force Coef/Deriv (1/deg), Along -X, {Cao,Ca_alf,PCa/PV,PCa/Ph,Ca_alfdot,Ca_q,Ca_bet}	:	0.63	0.001	0.0000	0.0000
Aero Force Coeff/Derivat (1/deg), Along Y, {Cyo,Cy_bet,Cy_r,Cy_alf,Cy_p,Cy_betdot,Cy_V}	:	0.0	-0.06	0.0000	0.0000
Aero Force Coeff/Deriv (1/deg), Along Z, {Czo,Cz_alf,Cz_q,Cz_bet,PCz/Ph,Cz_alfdot,PCz/PV}	:	-1.1	-0.31	0.0000	0.0000
Aero Moment Coeff/Derivat (1/deg), Roll: {Clo, Cl_beta, Cl_betdot, Cl_p, Cl_r, Cl_alfa}	:	0.0	-0.005	0.0000	0.0000
Aero Moment Coeff/Deriv (1/deg), Pitch: {Cmo,Cm_alfa,Cm_alfdot,Cm_bet,Cm_q,PCm/PV,PCm/Ph}	:	2.3	0.6	0.0000	0.0000
Aero Moment Coeff/Derivat (1/deg), Yaw : {Cno, Cn_beta, Cn_betdot, Cn_p, Cn_r, Cn_alfa}	:	0.0	-0.26	0.0000	0.0000

Number of Control Surfaces, With or No TWD (Tail-Wags-Dog and Hinge Moment Dynamics) ? : 0 NO TWD
Number of Thruster Engines, Include or Not the Tail-Wags-Dog and Load-Torque Dynamics ? : 2 NO TWD

TVC Engine No: 1	(Gimbaling Throttling Single_Gimbal)	: Left TVC	Gimbaling	
Engine Nominal Thrust, and Maximum Thrust in (lb) (for throttling)	:	320000.0	320000.0	
Mounting Angles wrt Vehicle (Dyn,Dzn), Maximum Deflections from Mount (Dymax,Dzmax) (deg):		0.5	0.0	5.0
Eng Mass (slug), Inertia about Gimbal (lb-sec ² -ft), Moment Arm, engine CG to gimbal (ft):		47.0	260.00	0.2
Gimbal location with respect to the Vehicle Reference Axes, Xgimb, Ygimb, Zgimb, in (ft) :		-144.6	-8.7	0.0
TVC Engine No: 2	(Gimbaling Throttling Single_Gimbal)	: Right TVC	Gimbaling	
Engine Nominal Thrust, and Maximum Thrust in (lb) (for throttling)	:	320000.0	320000.0	
Mounting Angles wrt Vehicle (Dyn,Dzn), Maximum Deflections from Mount (Dymax,Dzmax) (deg):		0.5	0.0	5.0
Eng Mass (slug), Inertia about Gimbal (lb-sec ² -ft), Moment Arm, engine CG to gimbal (ft):		47.0	260.00	0.2
Gimbal location with respect to the Vehicle Reference Axes, Xgimb, Ygimb, Zgimb, in (ft) :		-144.6	8.7	0.0

Number of Gyros, (Attitude and Rate) : 0
Number of Accelerometers, Along Axis: (x,y,z) : 0
Number of Vane Sensors, (Measuring Alpha or Beta) : 0
Number of Slosh Modes : 0
Number of Bending Modes : 0

CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)

Zulu, Stage-1, Max-Q, Rigid Body, Pitch Axis

Zulu, Stage-1, Max-Q, Rigid Body

! Pitch rigid system is extracted from the coupled rigid system above

TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS

Extract Inputs : 1 2 5
Extract States : 3 4 7
Extract Outputs: 3 4 7 14

FLIGHT VEHICLE INPUT DATA

Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh

! This is a Launch Vehicle during First Stage at Maximum Dynamic Pressure. It is launched from
! a high altitude to maximize orbit performance. At Max-Q and while flying at 2.5 Mach it
! experiences a strong wind shear that causes 4 deg of angle of attack. The vehicle has two side
! liquid oxygen rockets that can provide 320,000 (lb) of thrust each. The two fuel tanks, one
! on each side cause a significant amount of slosh disturbance.

! Body Axes Output,Attitude=Euler Angles,Without GAFD, No Turn Coordination

Vehicle Mass (lb-sec ² /ft), Gravity Accelerat. (g) (ft/sec ²), Earth Radius (Re) (ft)	:	8878.0	32.174	0.20896E+08	
Moments and products of Inertias Ixx, Iyy, Izz, Ixy, Ixz, Iyz, in (lb-sec ² -ft)	:	519500.0	1671000.0	2110000.0	0.0000
CG location with respect to the Vehicle Reference Point, Xcg, Ycg, Zcg, in (feet)	:	-126.05	0.0000	0.007	
Vehicle Mach Number, Velocity Vo (ft/sec), Dynamic Pressure (psf), Altitude (feet)	:	2.5	2000.0	220.00	83000.
Inertial Acceleration Vo_dot, Sensed Body Axes Accelerations Ax,Ay,Az (ft/sec ²)	:	62.000	54.000	0.0000	21.000
Angles of Attack and Sideslip (deg), alpha, beta rates (deg/sec)	:	4.0000	0.0000	0.0000	0.0000
Vehicle Attitude Euler Angles, Phi_o,Thet_o,Psi_o (deg), Body Rates Po,Qo,Ro (deg/sec)	:	0.0000	44.000	0.0000	0.0000
Wind Gust Vel wrt Vehi (Azim & Elev) angles (deg), or Force(lb), Torque(ft-lb), locat:xyz:			Gust	45.0	45.0
Surface Reference Area (feet ²), Mean Aerodynamic Chord (ft), Wing Span in (feet)	:	47.3	7.7000	7.7000	
Aero Moment Reference Center (Xmrc,Ymrc,Zmrc) Location in (ft), {Partial_rho/ Partial_H}	:	-127.2	0.0000	0.0000	0.0000
Aero Force Coef/Deriv (1/deg), Along -X, {Cao,Ca_alf,PCa/PV,PCa/Ph,Ca_alfdot,Ca_q,Ca_bet}	:	0.63	0.001	0.0000	0.0000
Aero Force Coeff/Derivat (1/deg), Along Y, {Cyo,Cy_bet,Cy_r,Cy_alf,Cy_p,Cy_betdot,Cy_V}	:	0.0	-0.06	0.0000	0.0000
Aero Force Coeff/Deriv (1/deg), Along Z, {Czo,Cz_alf,Cz_q,Cz_bet,PCz/Ph,Cz_alfdot,PCz/PV}	:	-1.1	-0.31	0.0000	0.0000
Aero Moment Coeff/Derivat (1/deg), Roll: {Clo, Cl_beta, Cl_betdot, Cl_p, Cl_r, Cl_alfa}	:	0.0	-0.005	0.0000	0.0000
Aero Moment Coeff/Deriv (1/deg), Pitch: {Cmo,Cm_alfa,Cm_alfdot,Cm_bet,Cm_q,PCm/PV,PCm/Ph}	:	2.3	0.6	0.0000	0.0000
Aero Moment Coeff/Derivat (1/deg), Yaw : {Cno, Cn_beta, Cn_betdot, Cn_p, Cn_r, Cn_alfa}	:	0.0	-0.26	0.0000	0.0000

Number of Control Surfaces, With or No TWD (Tail-Wags-Dog and Hinge Moment Dynamics) ? : 0 NO TWD
Number of Thruster Engines, Include or Not the Tail-Wags-Dog and Load-Torque Dynamics ? : 2 WITH TWD

TVC Engine No: 1	(Gimbaling Throttling Single_Gimbal)	: Left TVC	Gimbaling	
Engine Nominal Thrust, and Maximum Thrust in (lb) (for throttling)	:	320000.0	320000.0	
Mounting Angles wrt Vehicle (Dyn,Dzn), Maximum Deflections from Mount (Dymax,Dzmax) (deg):		0.5	0.0	5.0
Eng Mass (slug), Inertia about Gimbal (lb-sec ² -ft), Moment Arm, engine CG to gimbal (ft):		47.0	260.00	0.2
Gimbal location with respect to the Vehicle Reference Axes, Xgimb, Ygimb, Zgimb, in (ft) :		-144.6	-8.7	0.0
TVC Engine No: 2	(Gimbaling Throttling Single_Gimbal)	: Right TVC	Gimbaling	
Engine Nominal Thrust, and Maximum Thrust in (lb) (for throttling)	:	320000.0	320000.0	
Mounting Angles wrt Vehicle (Dyn,Dzn), Maximum Deflections from Mount (Dymax,Dzmax) (deg):		0.5	0.0	5.0
Eng Mass (slug), Inertia about Gimbal (lb-sec ² -ft), Moment Arm, engine CG to gimbal (ft):		47.0	260.00	0.2
Gimbal location with respect to the Vehicle Reference Axes, Xgimb, Ygimb, Zgimb, in (ft) :		-144.6	8.7	0.0

```

Number of Gyros, (Attitude and Rate) : 3
Gyro No 1 Axis:(Pitch,Yaw,Roll), (Attitude, Rate, Accelerat), Sensor Location in (feet) : Roll Rate -102.00 0.00 0.00
Gyro No 2 Axis:(Pitch,Yaw,Roll), (Attitude, Rate, Accelerat), Sensor Location in (feet) : Pitch Rate -102.00 0.00 0.00
Gyro No 3 Axis:(Pitch,Yaw,Roll), (Attitude, Rate, Accelerat), Sensor Location in (feet) : Yaw Rate -102.00 0.00 0.00

```

```

Number of Accelerometers, Along Axis: (x,y,z) : 2
Acceleromet No 1 Axis:(X,Y,Z), (Position, Velocity, Acceleration), Sensor Location (ft) : Y-axis Accelerat. -102.00 0.00 0.00
Acceleromet No 2 Axis:(X,Y,Z), (Position, Velocity, Acceleration), Sensor Location (ft) : Z-axis Accelerat. -102.00 0.00 0.00

```

```

Number of Vane Sensors, (Measuring Alpha or Beta) : 0

```

```

Number of Slosh Modes : 2
Left Tank (slug), Frequ Wy,Wz at lg (rad/s), Damping (zeta-y-z), Locat {Xsl,Ysl,Zsl} (ft): 350.0 2.31 2.31 0.001 0.001 -135.00 -8.7
Rght Tank (slug), Frequ Wy,Wz at lg (rad/s), Damping (zeta-y-z), Locat {Xsl,Ysl,Zsl} (ft): 350.0 2.31 2.31 0.001 0.001 -135.00 8.7

```

```

Number of Bending Modes : 8

```

```

Zulu, Stage-1, Max-Q, 8 Flex Modes, Pitch Modes

```

```

-----
SELECTED MODAL DATA AND LOCATIONS FOR THE PITCH MODES

```

```

Zulu, Stage-1, Max-Q, 8 Flex Modes, Pitch Modes

```

```

! The pitch modes were selected by applying +Z forces at the two gimbals, (same direction at the
! left and right TVC). The pitch rotation was sensed at the IMU node. We included also modal data
! at the two TVC gimbals, the 3 roll, pitch, and yaw gyros, and the two (Ny and Nz) acceleromet.
! The modal data were scaled from the original Nastran output in (*.mod) file. We reverse the
! directions in the X and Z axes. The modal slopes were multiplied by 12 to be converted from
! (rad/inch) to (radians/foot). The modal masses were multiplied by 12 to be converted from
! (snails) to (slugs). Standard Nastran to GN&C units transformation (Default).

```

```

MODE # 1/ 1, Frequency (rad/sec), Damping (zeta), Generalized Mass= 20.449 0.50000E-02 12.000
DEFINITION OF LOCATIONS (NODES) phi along X phi along Y phi along Z sigm about X sigm about Y sigm about Z

```

```

Node ID# Modal Data at the 2 Engines, (x,y,z)...
Left (-y) Boster Gimbal Locat 35701 -0.25060D-01 0.10089D-01 -0.48000D-03 0.23400D-04 -0.24816D-04 -0.12228D-02
Right (+y) Boster Gimbal Locat 45701 0.25080D-01 0.10093D-01 -0.98000D-04 0.22764D-04 -0.21000D-04 -0.12233D-02

```

```

Node ID# Modal Data at the 3 Gyros ...
Nav Base (IMU , Accelerom) 95602 -0.13080D-05 0.23100D-01 0.18520D-03 -0.60120D-04 0.17244D-04 0.36600D-02
Nav Base (IMU , Accelerom) 95602 -0.13080D-05 0.23100D-01 0.18520D-03 -0.60120D-04 0.17244D-04 0.36600D-02
Nav Base (IMU , Accelerom) 95602 -0.13080D-05 0.23100D-01 0.18520D-03 -0.60120D-04 0.17244D-04 0.36600D-02

```

```

Node ID# Modal Data at the 2 Accelerometers, along (x,y,z)...
Nav Base (IMU , Accelerom) 95602 -0.13080D-05 0.23100D-01 0.18520D-03
Nav Base (IMU , Accelerom) 95602 -0.13080D-05 0.23100D-01 0.18520D-03

```

```

Node ID# Modal Data at the 2 Slosh Masses...
Left Booster Liquid Oxygen S1 81602 -0.90090D-02 0.51000D-02 0.26000D-03 -0.33480D-05 0.24360D-04 0.26904D-02
Right Booster Liquid Oxygen S1 88603 0.90170D-02 0.50960D-02 0.34470D-03 -0.40680D-05 -0.83160D-05 0.26904D-02

```

```

Node ID# Modal Data at the Disturbance Point
Nav Base (IMU , Accelerom) 95602 -0.13080D-05 0.23100D-01 0.18520D-03 -0.60120D-04 0.17244D-04 0.36600D-02

```

```

MODE # 2/ 2, Frequency (rad/sec), Damping (zeta), Generalized Mass= 22.271 0.50000E-02 12.000
DEFINITION OF LOCATIONS (NODES) phi along X phi along Y phi along Z sigm about X sigm about Y sigm about Z

```

```

Node ID# Modal Data at the 2 Engines, (x,y,z)...
Left (-y) Boster Gimbal Locat 35701 -0.26580D-03 0.57370D-04 0.25310D-01 0.12300D-03 -0.19884D-02 -0.15852D-04
Right (+y) Boster Gimbal Locat 45701 0.41850D-03 0.88360D-04 0.25260D-01 0.11760D-03 -0.19872D-02 -0.18912D-04

```

```

Node ID# Modal Data at the 3 Gyros ...
Nav Base (IMU , Accelerom) 95602 0.43500D-03 0.45400D-03 -0.10500D-01 0.43200D-05 -0.23088D-02 0.54600D-04
Nav Base (IMU , Accelerom) 95602 0.43500D-03 0.45400D-03 -0.10500D-01 0.43200D-05 -0.23088D-02 0.54600D-04
Nav Base (IMU , Accelerom) 95602 0.43500D-03 0.45400D-03 -0.10500D-01 0.43200D-05 -0.23088D-02 0.54600D-04

```

```

Node ID# Modal Data at the 2 Accelerometers, along (x,y,z)...
Nav Base (IMU , Accelerom) 95602 0.43500D-03 0.45400D-03 -0.10500D-01
Nav Base (IMU , Accelerom) 95602 0.43500D-03 0.45400D-03 -0.10500D-01

```

```

Node ID# Modal Data at the 2 Slosh Masses...
Left Booster Liquid Oxygen S1 81602 -0.12800D-03 0.12550D-03 -0.24500D-01 -0.32760D-04 -0.11892D-02 0.13236D-04
Right Booster Liquid Oxygen S1 88603 0.15130D-03 0.28260D-03 -0.24510D-01 0.37200D-04 -0.81360D-03 0.70200D-04

```

```

Node ID# Modal Data at the Disturbance Point
Nav Base (IMU , Accelerom) 95602 0.43500D-03 0.45400D-03 -0.10500D-01 0.43200D-05 -0.23088D-02 0.54600D-04

```

```

-----
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)

```

```

Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh, Pitch Axis

```

```

Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh

```

```

! Pitch flexible system is extracted from the coupled system above
! to be used in Matlab/Simulink Simulations

```

```

TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS

```

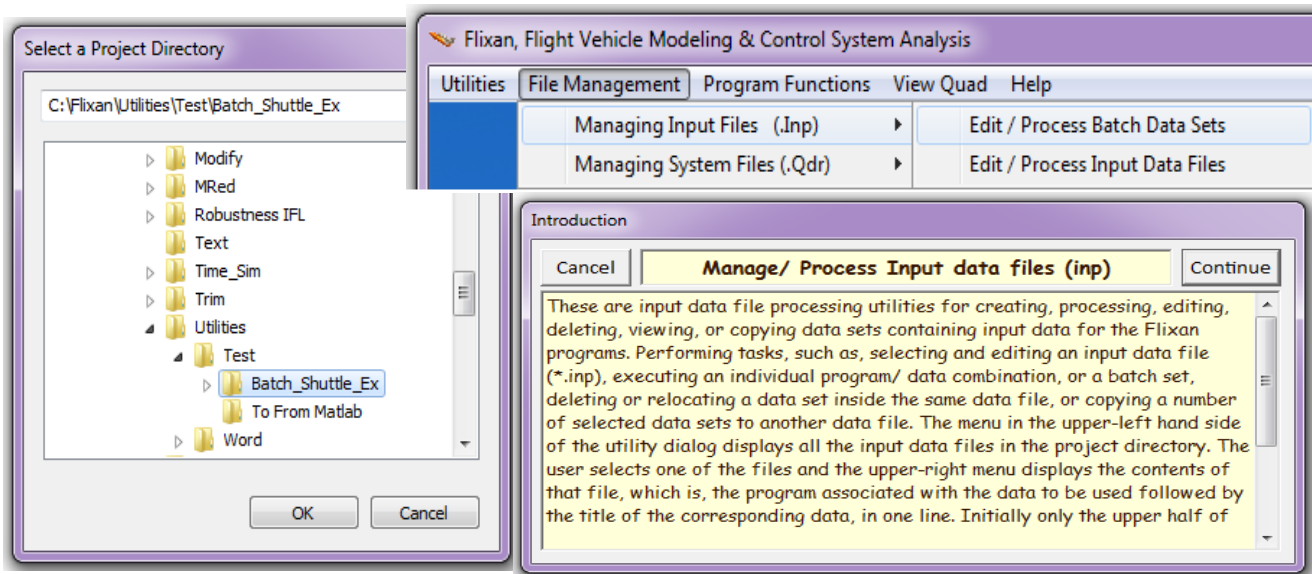
```

Extract Inputs : 1 2 3 4 9
Extract States : 3 4 7 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 31 33
Extract Outputs: 3 4 7 13 16 17 18

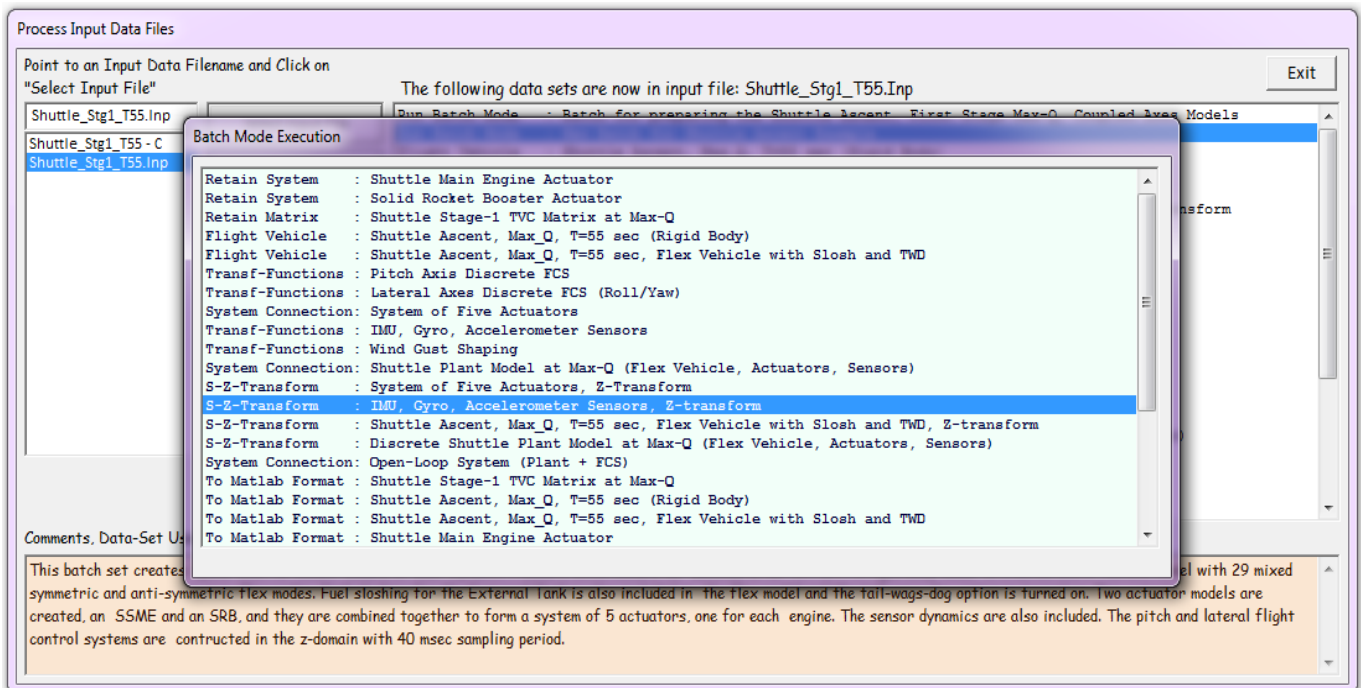
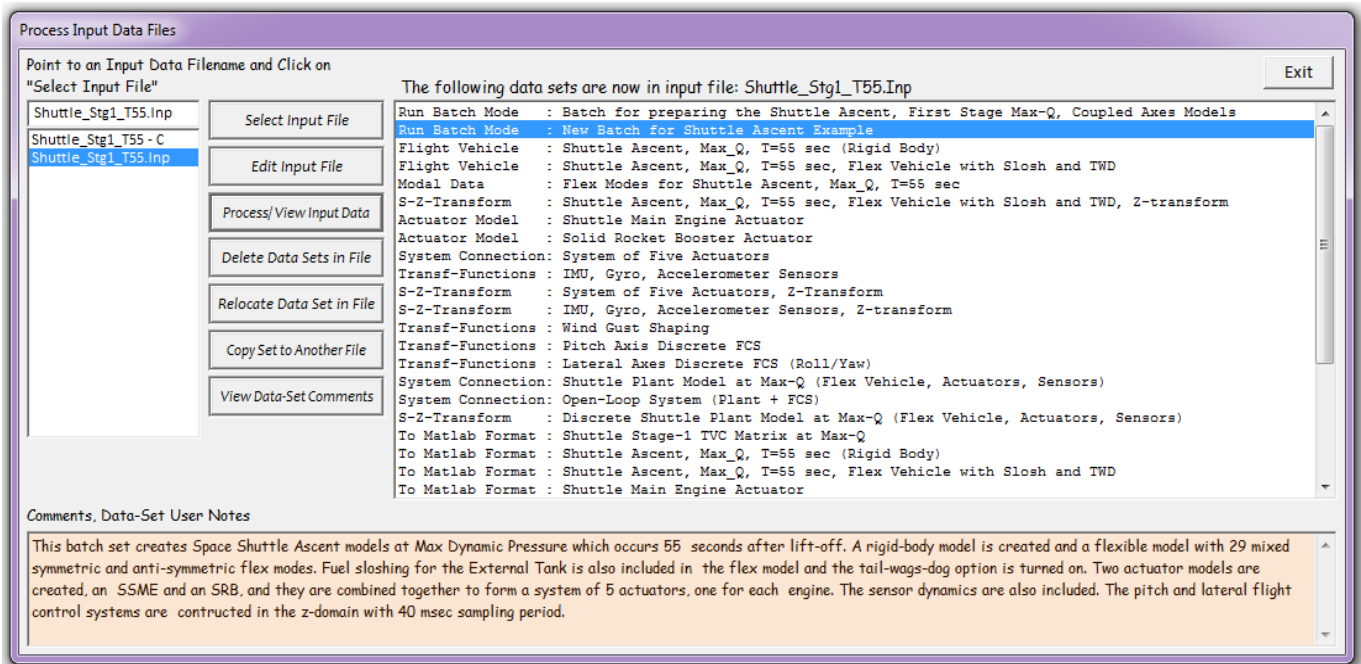
```

6.1.2 Utility for Creating, Editing and Managing Batch Data-Sets

Batch sets are important for quickly processing input data files and a batch set utility is included in Flixan for creating and managing batch sets. A batch is a set of input instructions that coordinate non-interactive processing of data-sets within an input file by Flixan utilities. This is in contrast with the interactive processing that uses menus, dialogs, and graphics. The batch manager is a utility that is not only used to create new batch sets but also to modify existing data-sets and to save them under a different title in the input data file (.Inp). Batch sets are typically created for future applications after the modeling process has already been debugged interactively. To create a set of batch commands you must first plan the data processing sequence and debug one process at a time by running each data-set/utility combination interactively, to make sure that they can be processed successfully without any errors caused by erroneous input data. When the individual steps are successfully tested, then you can use the batch utility to create a batch that will process the individual steps faster. The following is an example of creating and running a batch using the batch editor. You must first select the directory that includes at least one input data file (.Inp). In this case choose folder: "C:\Flixan\Utilities\Test\Batch_Shuttle_Ex". Then go to Flixan main menu, click on "File Management", then "Manage Input Files", and "Edit/ Process Batch Data Sets".



This will open the batch editor dialog shown below. From the filename selection menu on the left side select the input file "Shuttle_Stg1_T55.Inp" and click "Select Input File". This input file already includes previously created data-sets ready to be processed by Flixan programs. The individual sets are either created interactively by Flixan or copied from other input files using a text editor. The menu at the bottom of this dialog shows the titles of all the data sets which are included in this file. The corresponding utility program is also shown on the left side of each title. At this point the program expects the user to either create a new batch or to modify an already existing batch. In this case the file already contains a batch set and the editor prompts the user to either select the existing batch title from a menu or to create a new batch. In this case we respond with "Create New" batch. We must then enter a new title for the new batch to be created: "New Batch for the Shuttle Ascent Example". The light blue menu at the top of the dialog will eventually show the accumulated batch commands that will be included in this batch and processed by the batch processor. Titles from the yellow menu at the bottom are gradually selected and included in the batch, one or more items at a time by clicking on "Include Title in Batch Set".



The new batch is then processed by Flixan and the new systems are saved in file: "*Shuttle_Stg1-T55.Inp*".

Definition of System Variables

```
Inputs = 2
  Nozzle Deflection Command (Delta_com) (rad)
  Load-Torque at the Gimbal, (Tl) (ft-lb)

Outputs = 3
  Nozzle Gimbal Rotation (Delta) (rad)
  Nozzle Gimbal Rate (Delta-dot) (rad/sec)
  Gimbal Accelerat. (Delta-dot-dot) (rad/sec^2)

States = 4
  Differ. Pressure Feedback (DPF) (ft-lb)
  Piston Displacement (Xr) (feet)
  Engine Gimbal Rotation (Delta) (rad)
  Engine Gimbal Rate (Delta-dot) (rad/sec)
```

Gain Matrix for ...

Pitch axis Mixing Matrix for Rocket-Plane [Kpitmix: Q-dot, Ax]

! Mixing Logic Matrix for the rocket-plane in the pitch axis. It converts the pitch and
! axial acceleration demands to five aero-surface deflections and engine thrust variation

```
Matrix Kpitmix          Size = 6 X 2
  1-Column          2-Column
1-Row -0.488779040922E-01  0.982582014775E-01
2-Row -0.291806701836E-01 -0.310962189063E-02
3-Row -0.291806701836E-01 -0.310962189063E-02
4-Row -0.127295123699E+00  0.117463289448E-03
5-Row -0.127295123699E+00  0.117463289448E-03
6-Row -0.370557557847E-01  0.137897247667E-02
```

Definitions of Matrix Inputs (Columns): 2
Q-dot Pitch Accel Demand About Y Axis
Ax Forward Accel Demand Along X Axis

Definitions of Matrix Outputs (Rows): 6
Output: 1 Delta(throttle) of Engine No: 1
Output: 2 Control Surface: 1 Deflection (radians)
Output: 3 Control Surface: 2 Deflection (radians)
Output: 4 Control Surface: 3 Deflection (radians)
Output: 5 Control Surface: 4 Deflection (radians)
Output: 6 Control Surface: 5 Deflection (radians)

STATE-SPACE SYSTEM ...

First Order Gust Lag

! A low-pass filter 0.5 (rad/sec) to smooth out the input gust
Number of Inputs, States, Outputs, Sample Time dT (for discrete)= 1 1 1 0.0000
Matrices: (A,B,C,D)

```
Matrix A          Size = 1 X 1
  1-Column
1-Row -0.500000000000E+00
```

Matrix B Size = 1 X 1
 1-Column
1-Row 0.100000000000E+01

Matrix C Size = 1 X 1
 1-Column
1-Row 0.500000000000E+00

Matrix D Size = 1 X 1
 1-Column
1-Row 0.000000000000E+00

Definition of System Variables

```
Inputs = 1
  1 Input No: 1

States = 1
  1 State No: 1

Outputs = 1
  1 Output No: 1
```

STATE-SPACE SYSTEM ...

Pitch Design Model

! The pitch axis control design model is extracted from the coupled rigid-body model. The ! wind-gust input is not used. The states and outputs are: (theta, q, and alpha)

Number of Inputs, States, Outputs, Sample Time dT (for discrete)= 6 3 3 0.0000

Matrices: (A,B,C,D)

Matrix A Size = 3 X 3
1-Column 2-Column 3-Column
1-Row 0.000000000000E+00 0.999996066093E+00 0.000000000000E+00
2-Row 0.000000000000E+00 -0.301385824515E+00 -0.398388623235E+01
3-Row 0.533570410077E-08 0.997318761156E+00 -0.303460356762E+00

Matrix B Size = 3 X 6
1-Column 2-Column 3-Column 4-Column 5-Column
6-Column
1-Row 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
2-Row -0.383726824856E-01 -0.873361046148E+00 -0.873361046148E+00 -0.363483208593E+01 -
3-Row -0.149738604708E-03 -0.251044848966E-01 -0.251044848966E-01 -0.214350479379E-01 -

Matrix C Size = 3 X 3
1-Column 2-Column 3-Column
1-Row 0.100000000000E+01 0.000000000000E+00 0.000000000000E+00
2-Row 0.000000000000E+00 0.100000000000E+01 0.000000000000E+00
3-Row 0.000000000000E+00 0.000000000000E+00 0.100000000000E+01

Matrix D Size = 3 X 6
1-Column 2-Column 3-Column 4-Column 5-Column
6-Column
1-Row 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
2-Row 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
3-Row 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00

Definition of System Variables

Inputs = 6
1 Throttle Input dTh/Th for Engine No 1 (-)
2 Aero Surface No 1 Deflection (radians)
3 Aero Surface No 2 Deflection (radians)
4 Aero Surface No 3 Deflection (radians)
5 Aero Surface No 4 Deflection (radians)
6 Aero Surface No 5 Deflection (radians)

States = 3
1 Pitch Attitude (theta-rigid) (radians)
2 Pitch Rate (q-rigid) (rad/sec)
3 Angle of attack (alfa-rigid) (radians)

Outputs = 3
1 Pitch Attitude (thet-123) (radians)
2 Pitch Rate (q-body) (rad/sec)
3 Angle of attack, alfa, (radians)

STATE-SPACE SYSTEM ...

Pitch Design Model with Alpha Integral (2-inputs, 4-states)

! The pitch axis control design model is augmented with the integral of the angle of attack for ! H-infinity design. The states and outputs are: (theta, q, alpha, and alpha-integral). The ! inputs to the design model are reduced to two by pre-multiplying it with the mixing matrix ! (Kpitmix)

Number of Inputs, States, Outputs, Sample Time dT (for discrete)= 2 4 4 0.0000

Matrices: (A,B,C,D)

Matrix A Size = 4 X 4
1-Column 2-Column 3-Column 4-Column
1-Row 0.000000000000E+00 0.999996066093E+00 0.000000000000E+00 0.000000000000E+00
2-Row 0.000000000000E+00 -0.301385824515E+00 -0.398388623235E+01 0.000000000000E+00
3-Row 0.533570410077E-08 0.997318761156E+00 -0.303460356762E+00 0.000000000000E+00
4-Row 0.000000000000E+00 0.000000000000E+00 0.100000000000E+01 0.000000000000E+00

Matrix B Size = 4 X 2
1-Column 2-Column
1-Row 0.000000000000E+00 0.000000000000E+00
2-Row 0.999912364254E+00 0.751375398850E-06
3-Row 0.702100385268E-02 0.132980916263E-03
4-Row 0.000000000000E+00 0.000000000000E+00

```

Matrix C                               Size = 4 X 4
      1-Column      2-Column      3-Column      4-Column
1-Row 0.10000000000000E+01  0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
2-Row 0.00000000000000E+00  0.10000000000000E+01  0.00000000000000E+00  0.00000000000000E+00
3-Row 0.00000000000000E+00  0.00000000000000E+00  0.10000000000000E+01  0.00000000000000E+00
4-Row 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00  0.10000000000000E+01
-----

```

```

Matrix D                               Size = 4 X 2
      1-Column      2-Column
1-Row 0.00000000000000E+00  0.00000000000000E+00
2-Row 0.00000000000000E+00  0.00000000000000E+00
3-Row 0.00000000000000E+00  0.00000000000000E+00
4-Row 0.00000000000000E+00  0.00000000000000E+00
-----

```

Definition of System Variables

```

Inputs = 2
  1 Pitch Acceleration
  2 Axial Acceleration

```

```

States = 4
  1 State No: 1
  2 State No: 2
  3 State No: 3
  4 State No: 4

```

```

Outputs = 4
  1 Pitch Attitude (theta-rigid) (radians)
  2 Pitch Rate (q-rigid) (rad/sec)
  3 Angle of attack (alpha) (radians)
  4 Alpha Integral (rad-sec)

```

STATE-SPACE SYSTEM ...

Pitch Design Model with Alpha Integral (1-input, 4-states)

! Eliminate the second input (Ax acceleration) from the pitch design model

Number of Inputs, States, Outputs, Sample Time dT (for discrete)= 1 4 4 0.0000

Matrices: (A,B,C,D)

```

Matrix A                               Size = 4 X 4
      1-Column      2-Column      3-Column      4-Column
1-Row 0.00000000000000E+00  0.999996066093E+00  0.00000000000000E+00  0.00000000000000E+00
2-Row 0.00000000000000E+00 -0.301385824515E+00 -0.398388623235E+01  0.00000000000000E+00
3-Row 0.533570410077E-08  0.997318761156E+00 -0.303460356762E+00  0.00000000000000E+00
4-Row 0.00000000000000E+00  0.00000000000000E+00  0.10000000000000E+01  0.00000000000000E+00
-----

```

```

Matrix B                               Size = 4 X 1
      1-Column
1-Row 0.00000000000000E+00
2-Row 0.999912364254E+00
3-Row 0.702100385268E-02
4-Row 0.00000000000000E+00
-----

```

```

Matrix C                               Size = 4 X 4
      1-Column      2-Column      3-Column      4-Column
1-Row 0.10000000000000E+01  0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
2-Row 0.00000000000000E+00  0.10000000000000E+01  0.00000000000000E+00  0.00000000000000E+00
3-Row 0.00000000000000E+00  0.00000000000000E+00  0.10000000000000E+01  0.00000000000000E+00
4-Row 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00  0.10000000000000E+01
-----

```

```

Matrix D                               Size = 4 X 1
      1-Column
1-Row 0.00000000000000E+00
2-Row 0.00000000000000E+00
3-Row 0.00000000000000E+00
4-Row 0.00000000000000E+00
-----

```

Definition of System Variables

```

Inputs = 1
  1 Pitch Acceleration

```

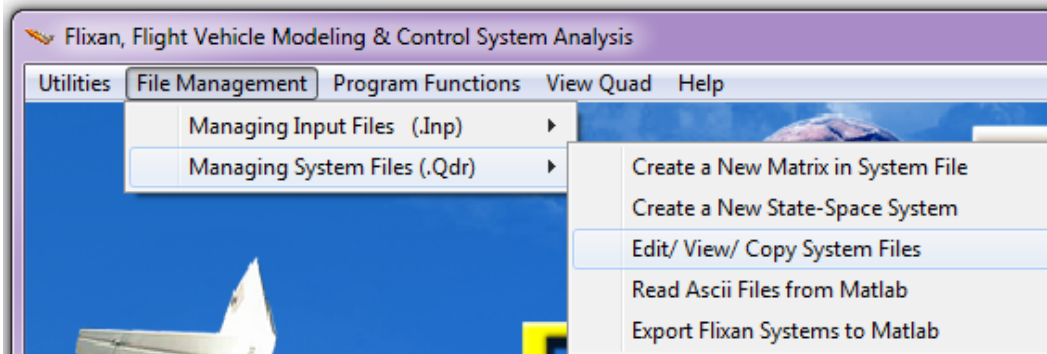
```

States = 4
  1 State No: 1
  2 State No: 2
  3 State No: 3

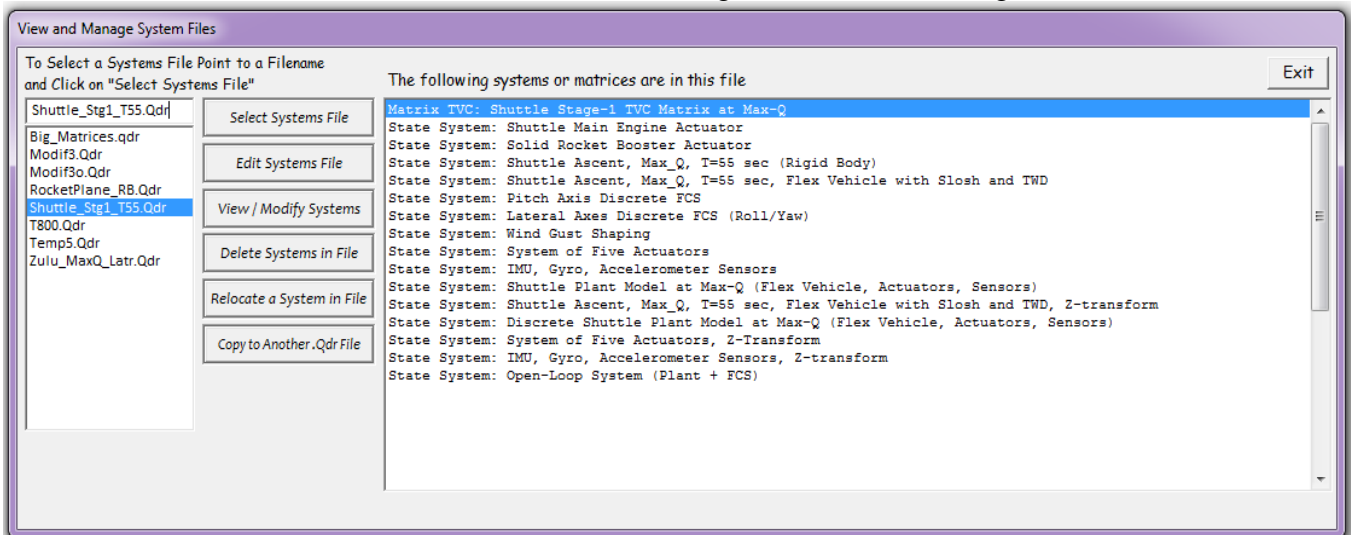
```

6.2.1 Managing System Files

The systems file management utility is used for organizing system files and it is similar to the input data file management utility. Maintaining organized system files by introducing comments, labeling the system inputs, states, and output variables, keeping the systems and matrices in a sequential order, and deleting the unused older versions is a very attractive feature for documenting the analysis. The comments, notes and name-lists of the system variables are useful for remembering details, especially when you have to discontinue the analysis for a period of time and resume it later in the future. To get to the systems manager utility, the user must first select the project directory, and from the Flixan main menu select: "*File Management*", "*Managing System Files (Qdr)*", and "*Edit/ View/ Copy System Files*", as shown below.



The systems file manager dialog is shown below. The menu on the left side shows the system files which are located in the project directory. The user selects a filename, for example, "*Shuttle_Stg1_T55.Qdr*", and clicks on "*Select Systems File*". The menu on the right will then display the titles of the systems or single matrices which are saved in the systems file, as shown below. The user is then able to perform several operations on this file using the dialog buttons. For example, delete some of the systems or matrices which are no longer used, move them to a different location within the file, and edit the systems file using a Flixan provided text editor. The repositioning of each item is defined relative to the top of the file. The "*View/ Modify Systems*" button is used to display the state-space system or matrix using color coded graphics, including tables of the system inputs, states, outputs, and display the user comments or notes that describe the system or the gain matrix. The state variable definitions and comments can be modified using the interactive dialogs.



system inputs, states, and output variables. There are similar tabs for single matrices. In addition to modifying the matrix elements, the variable definitions or comments can also be modified using this dialog. The new system, however, must be saved under a new title. The first system below is a lateral vehicle with four inputs and three outputs. The display shows the user comments. The second example shows a Shuttle main engine actuator. The display tab shows definitions of the state variables. The bottom display is pointing at matrix element B(1,1) that has a value of 40.22 .

$\dot{x} = Ax + Bu$
 $y = Cx + Du$

Sampling Period (dT) in (sec), Zero if System is Continuous:

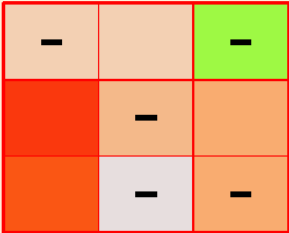
Inputs | **Outputs** | States | System Description

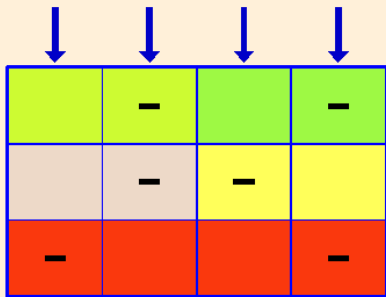
A further simplified lateral axes rigid-body model is extracted from the coupled rigid-body model to be used for LQR control design. It consists of three states: (p, r, and beta)

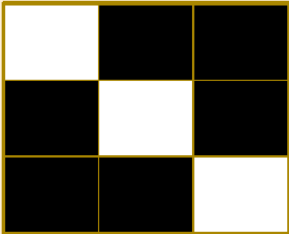
Select an Input, State or Output variable from the menu above and press Edit. Then you may type in a new description for that variable in the field above.
Repeat to change other Inputs, States, Outputs, or State-Space System Elements. Then click on "Save Changes" to save the new values and titles or Exit the dialog.

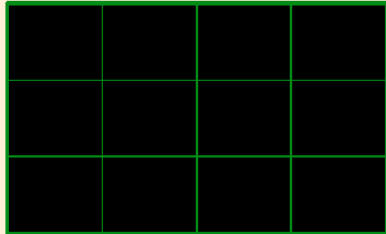
To Create a new System, Click on the Matrix Elements, Enter the new values and Click on "Save Changes"

Lateral Design Model
State-Space System Title









$\dot{x} = Ax + Bu$
 $y = Cx + Du$

Sampling Period (dT) in (sec), Zero if System is Continuous:

Inputs | **Outputs** | States | System Description


Differ. Pressure Feedback (DPF) (ft-lb)
Piston Displacement (Xr) (feet)
Engine Gimbal Rotation (Delta) (rad)
Engine Gimbal Rate (Delta-dot) (rad/sec)

Select an Input, State or Output variable from the menu above and press Edit. Then you may type in a new description for that variable in the field above.
Repeat to change other Inputs, States, Outputs, or State-Space System Elements. Then click on "Save Changes" to save the new values and titles or Exit the dialog.


Matrix B Element: (1, 1)=

To Create a new System, Click on the Matrix Elements, Enter the new values and Click on "Save Changes"

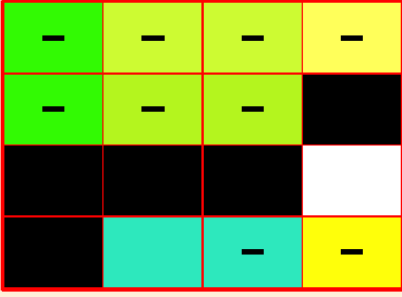
Color Code for Magnitudes between Zero (black) and One (white)

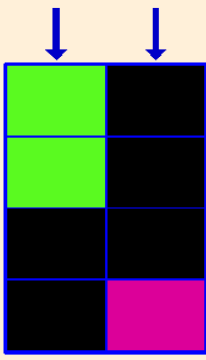


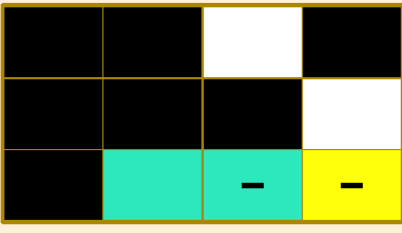
Color Code for Magnitudes between One (white) and Infinity (blue)

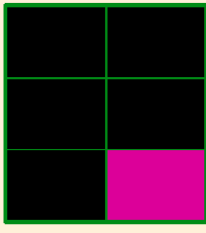


Shuttle Main Engine Actuator
State-Space System Title



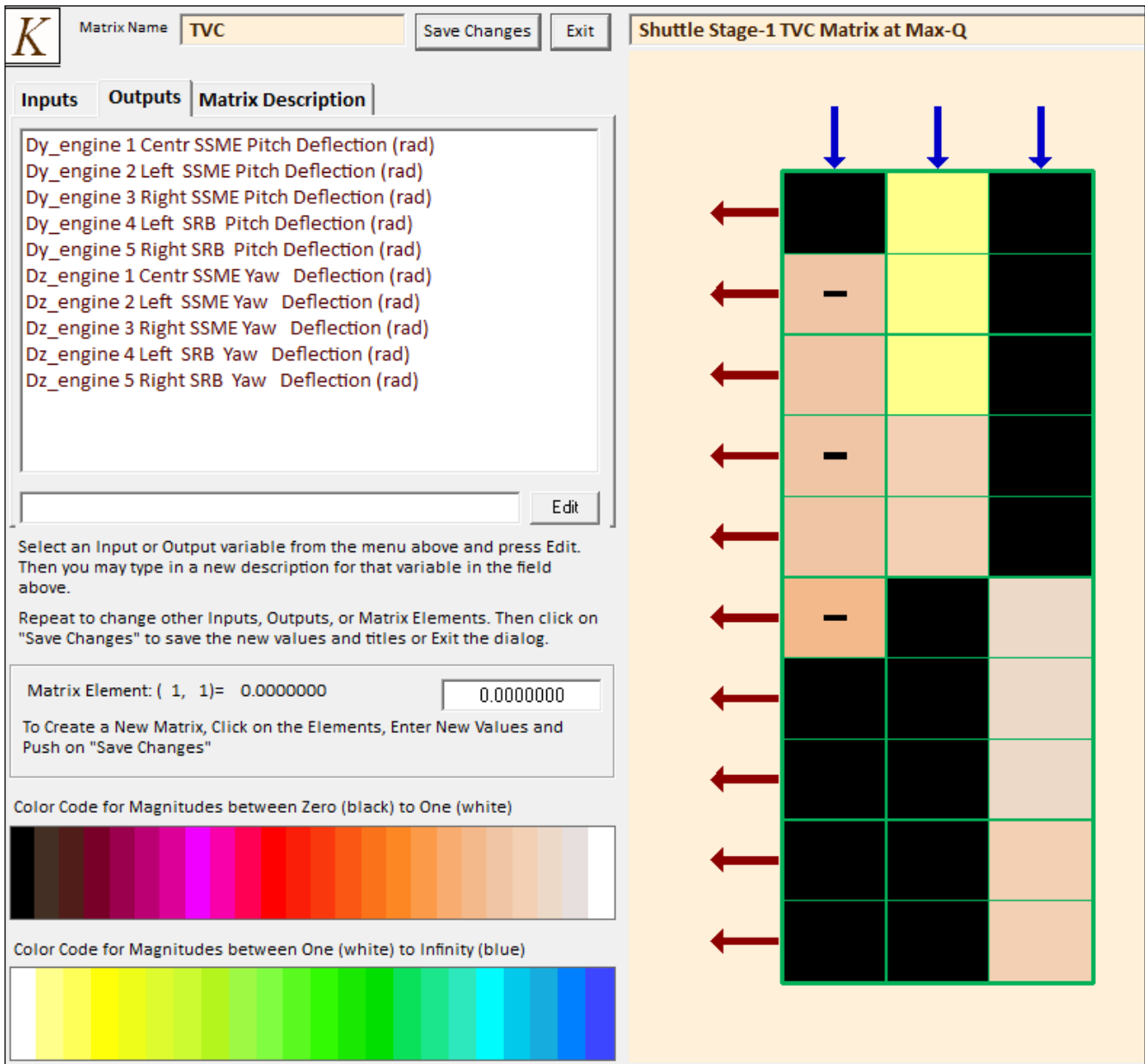




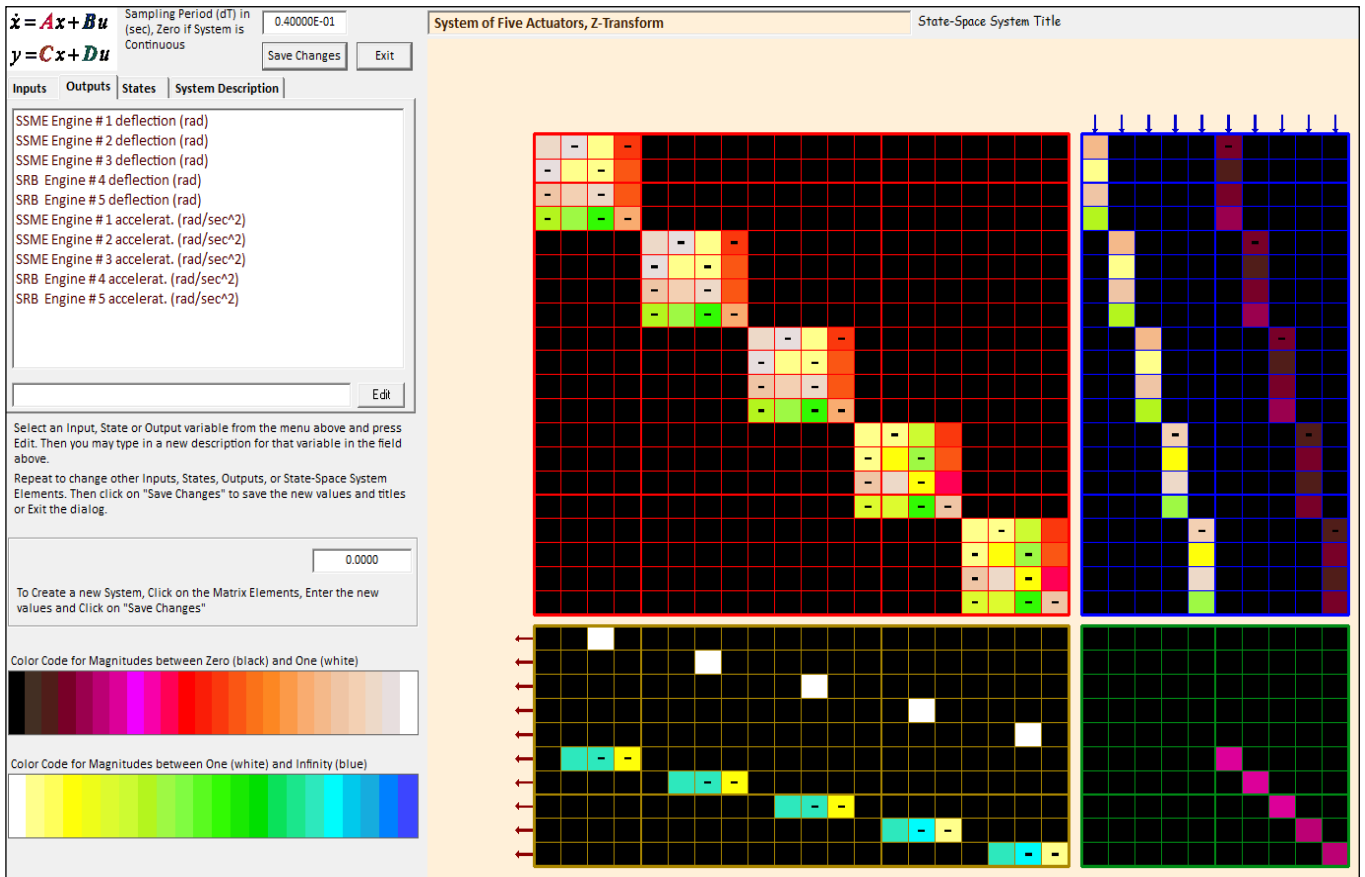
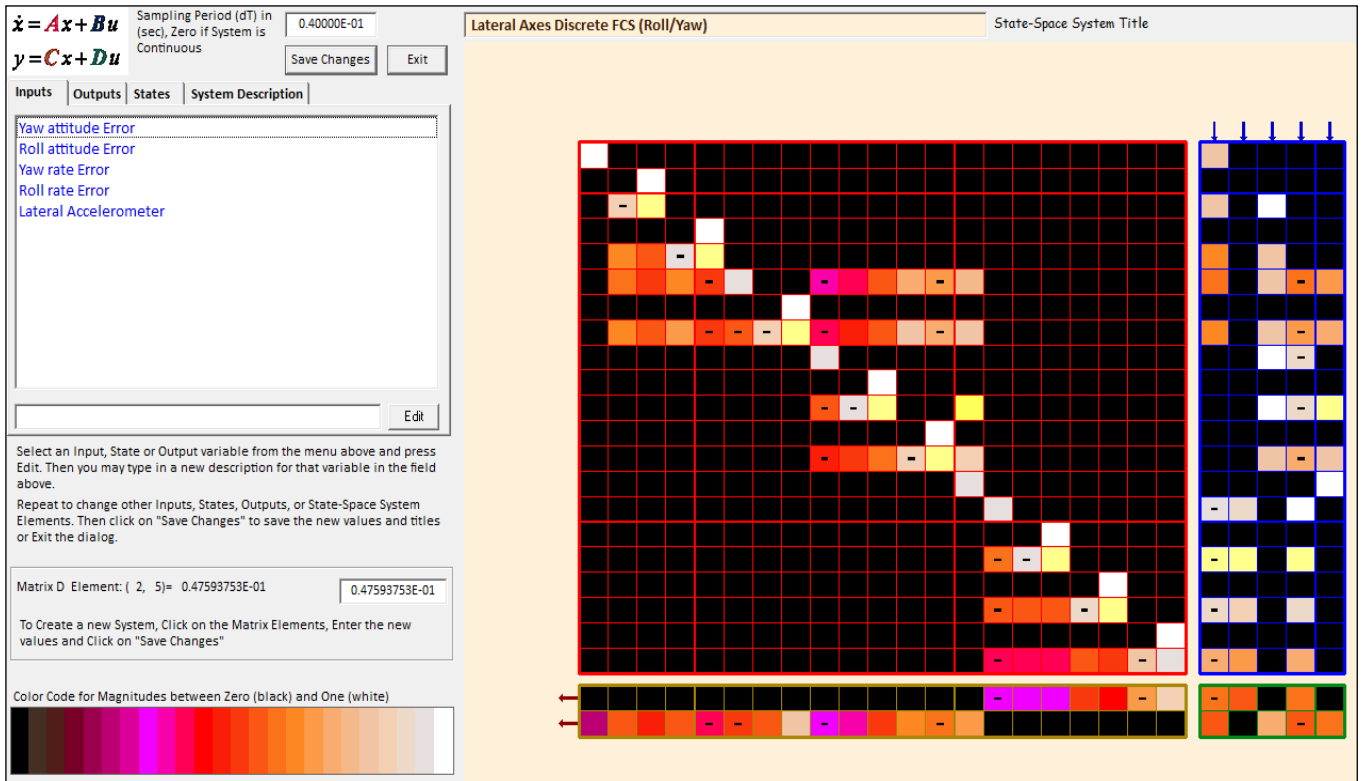


129

The figure below shows a Shuttle TVC matrix which transforms the roll, pitch and yaw acceleration demands to pitch and yaw deflections for the 5 engines. The matrix name and title are shown on the top and they can be modified and saved to a different name and title. Every time you change something using this dialog, for example, a matrix element, the input-output definitions, the comments, the matrix name or title, you must remember to save the changes by clicking the “*Save Changes*” button. When the modifications are complete, click on the “*Exit*” button, and the new data will be saved as a separate system or matrix in the same systems file. You must remember to change the system or matrix title, otherwise, the modifications will not be accepted using the same title. You can later delete the old system or matrix and rename the new one, if you wish.

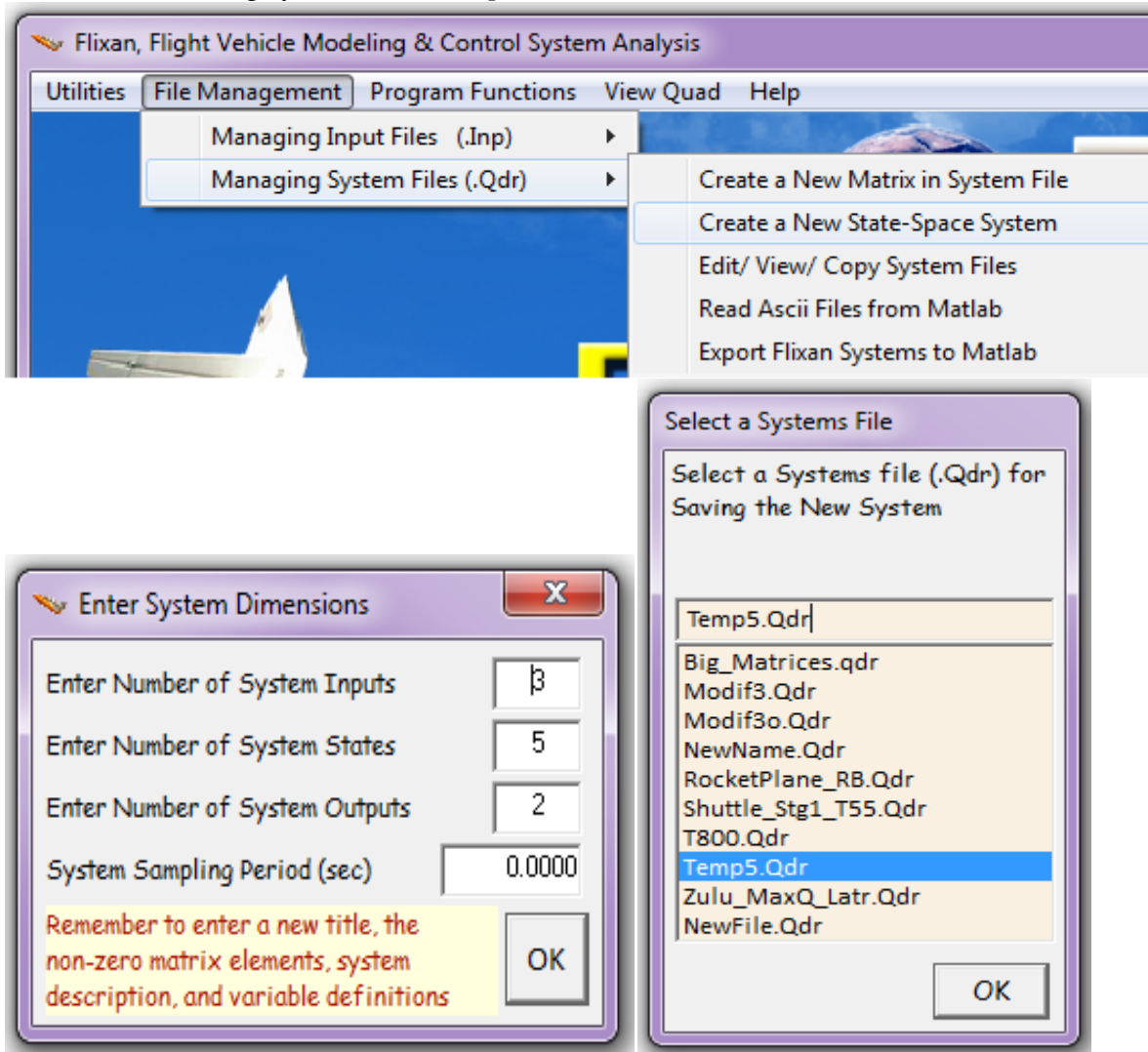


The following figures show two additional discrete systems sampled at 40 msec. A Shuttle lateral flight control system where the tab is showing its inputs, and a system of five actuators, three SSME and two SRB in block diagonal form. The second system's tab is showing the outputs.



6.2.2 Creating a New System or a Matrix

The Flixan program also includes utilities for creating a new state-space system or a new matrix from scratch by entering the data using dialogs and then saving the system or matrix in a systems file (.Qdr). After selecting the project directory, from the Flixan main menu select “File Management”, then “Manage System Files”, and then “Create a New State-Space System” or “Create a New Matrix”. For a new system the following dialog appears asking the user to define the number of the new system inputs, states, and outputs, and also the sampling period if the system is discrete described in terms of state-space difference equations. Otherwise, for a continuous system the sampling period is set to zero. In this case we arbitrarily define a system with 3 inputs, 2 outputs, and 5 states. The next menu shows the system files already existing in the current directory. A filename must be selected to save the new system or matrix. To create a new file the user must select “NewFile.Qdr”. In this example, however, we select an existing systems file “Temp5.Qdr”.



At this point a dialog appears that helps the user define the new system. Initially the four state-space matrices (A, B, C, D) are empty. The matrix element numbers are not yet entered and the element squares are black. In the upper hand left side of the dialog there a field with four tabs used for defining the new system's input, state, and output variables, and also for entering user comments about the system. The sizes of the four (A, B, C, D) matrices on the right are defined by the number of inputs,

states, and outputs. Their elements are initially set to zero (black) and they are ready to be painted by the user, one element at a time.

$\dot{x} = Ax + Bu$ Sampling Period (dT) in (sec, Zero if System is Continuous) 0.0000 Save Changes Exit

$y = Cx + Du$

Inputs Outputs States System Description

Select an Input, State or Output variable from the menu above and press Edit. Then you may type in a new description for that variable in the field above.

Repeat to change other Inputs, States, Outputs, or State-Space System Elements. Then click on "Save Changes" to save the new values and titles or Exit the dialog.

0.0000

To Create a new System, Click on the Matrix Elements, Enter the new values and Click on "Save Changes"

Color Code for Magnitudes between Zero (black) and One (white)

New System Title State-Space System Title

$\dot{x} = Ax + Bu$ Sampling Period (dT) in (sec, Zero if System is Continuous) 0.0000 Save Changes Exit

$y = Cx + Du$

Inputs Outputs States System Description

This is a user created system using the system creation dialog by typing the non-zero elements directly in the dialog. The user must first enter the size of the new system, enter the new system title, define the state variables, and a short paragraph that describes the new system (not more than 2000 characters). To define a matrix element, point the mouse cursor to a matrix element, type the element value in the field below, and click "Save Changes". The element will change color according to the new value.

Select an Input, State or Output variable from the menu above and press Edit. Then you may type in a new description for that variable in the field above.

Repeat to change other Inputs, States, Outputs, or State-Space System Elements. Then click on "Save Changes" to save the new values and titles or Exit the dialog.

Matrix D Element: (2, 3)= 0.0000000 0.6000000E-06

To Create a new System, Click on the Matrix Elements, Enter the new values and Click on "Save Changes"

Color Code for Magnitudes between Zero (black) and One (white)

Color Code for Magnitudes between One (white) and Infinity (blue)

User Created System State-Space System Title

The user must point the cursor at a matrix element, click the mouse button to select the element, enter the value of the selected element in the field below the tabs, and click on the "Save Changes" button.